A5M

dataday 1994

Memo

Information contained within this journal is provided as-is without warranty or claim for having suitability for any purpose what-so-ever and is provided for informational purposes only.

dataday 1994

Charlote Greenwood Programming Journal 1994

Dated between 1994 and 1995.
Hardcover book, A5 size, approx 132 pages (original)

ar Planner

	Jan	Feb	March	April	May	June
ay						
ay		1	1 ☐			
esday		2	2			1
day		3	3			2
		4	4	1 C		3
lay	1	5	5	2		4
y	2	6	6	3	1 C	5
ay	3	7	7	4	2	6
ay	4	8	8	5	3 ☒ 3·00	7
esday	5	9	9	6	4 10·00	8
day	6	10	10	7	5 3·35	9
	7	11	11	8	6	10
lay	8	12	12	9	7	11
y	9	13	13	10	8	12
ay	10	14	14	11	9	13
ay	11	15	15 10·30AM	12	10	14
esday	12	16	16 3·00PM	13	11	15
day	13	17	17 8AM	14	12	16
	14	18	18 11AM		13	17
day	15	19	19	16	14	18
y	16	20	20	17	15	19
lay	17	21	21 T- 12:00		16	20
ay	18	22 x ☒	22 x ☐ 1PM	20	17	21
esday	19	23	23		18	22
day	20	24	24	21	19	23
	21	25	25	22 x ☐	20	24
day	22	26	26	23	21	25
ay	23	27	27	24	22	26
lay	24	28	28	25	23 x ☐	27
lay	25		29	26	24	28
esday	26		30	27	25	29
day	27		31	28	26	30
y	28			29	27	
day	29			30	28	
ay	30				29	
lay	31				30	
lay					31	

9th THR 3.35

uly	Aug	Sept	Oct	Nov	Dec	
	1					Monday
	2			1		Tuesday
	3			2		Wednesday
	4	1		3	1	Thursday
	5	2		4	2	Friday
	6	3	1	5	3	Saturday
	7	4	2	6	4	Sunday
	8	5	3	7	5	Monday
	9	6	4	8	6	Tuesday
	10	7	5	9	7	Wednesday
7	11	8	6	10	8	Thursday
8	12	9	7	11	9	Friday
9	13	10	8	12	10	Saturday
10	14	11	9	13	11	Sunday
11	15	12	10	14	12	Monday
12	16	13	11	15	13	Tuesday
13	17	14	12	16	14	Wednesday
14	18	15	13	17	15	Thursday
15	19	16	14	18	16	Friday
16	20	17	15	19	17	Saturday
17	21	18	16	20	18	Sunday
18	22	19	17	21	19	Monday
19	23	20	18	22	20	Tuesday
20	24	21	19	23	21	Wednesday
21	25	22	20	24	22	Thursday
22	26	23	21	25	23	Friday
23	27	24	22	26	24	Saturday
24	28	25	23	27	25	Sunday
25	29	26	24	28	26	Monday
26	30	27	25	29	27	Tuesday
27	31	28	26	30	28	Wednesday
28		29	27		29	Thursday
29		30	28		30	Friday
30			29		31	Saturday
31			30			Sunday
			31			Monday
						Tuesday

1993 Bank & Public Holidays

May

Wk	M	T	W	T	F	S	S
17						1	2
18	3	4	5	6	7	8	9
19	10	11	12	13	14	15	16
20	17	18	19	20	21	22	23
21	24	25	26	27	28	29	30
22	31						

June

Wk	M	T	W	T	F	S	S
22		1	2	3	4	5	6
23	7	8	9	10	11	12	13
24	14	15	16	17	18	19	20
25	21	22	23	24	25	26	27
26	28	29	30				

November

Wk	M	T	W	T	F	S	S
44	1	2	3	4	5	6	7
45	8	9	10	11	12	13	14
46	15	16	17	18	19	20	21
47	22	23	24	25	26	27	28
48	29	30					

December

Wk	M	T	W	T	F	S	S
48			1	2	3	4	5
49	6	7	8	9	10	11	12
50	13	14	15	16	17	18	19
51	20	21	22	23	24	25	26
52	27	28	29	30	31		

England and Wales

New Year	January 1
Good Friday	April 9
Easter Monday	April 12
Bank Holiday	May 3
Bank Holiday	May 31
Bank Holiday	August 30
Christmas Day	December 25
Boxing Day (Sun)	December 26
Bank Holidays	December 27/28

Scotland

New Year	January 1
Bank Holiday	January 4
Good Friday	April 9
Bank Holiday	May 3
Bank Holiday	May 31
Bank Holiday	August 2
Christmas Day	December 25
Boxing Day (Sun)	December 26
Bank Holidays	December 27/28

Northern Ireland

New Year	January 1
St. Patrick's Day	March 17
Good Friday	April 9
Easter Monday	April 12
Bank Holiday	May 3
Bank Holiday	May 31
Bank Holiday	July 12
Bank Holiday	August 30
Christmas Day	December 25
Boxing Day (Sun)	December 26
Bank Holidays	December 27/28

Republic of Ireland

New Year	January 1
St. Patrick's Day	March 17
Good Friday	April 9
Easter Monday	April 12
Bank Holiday	June 7
Bank Holiday	August 2
Bank Holiday	October 25
Christmas Day	December 25
St. Stephen's Day (Sun)	Dec 26
Bank Holidays	Dec 27/28

1994 Bank & Public Holidays

May

Wk	M	T	W	T	F	S	S
17							1
18	2	3	4	5	6	7	8
19	9	10	11	12	13	14	15
20	16	17	18	19	20	21	22
21	23	24	25	26	27	28	29
22	30	31					

June

Wk	M	T	W	T	F	S	S
22			1	2	3	4	5
23	6	7	8	9	10	11	12
24	13	14	15	16	17	18	19
25	20	21	22	23	24	25	26
26	27	28	29	30			

November

Wk	M	T	W	T	F	S	S
44							
45							
46							
47							
48							

December

Wk	M	T	W	T	F	S	S
48							
49							
50							
51							
52							

England and Wales

New Year	January 3
Good Friday	April 1
Easter Monday	April 4
Bank Holiday	May 2
Bank Holiday	May 30
Bank Holiday	August 29
Christmas Day (Sun)	December 25
Boxing Day	December 26
Bank Holiday	December 27

Scotland

New Year	January 3
Bank Holiday	January 4
Good Friday	April 1
Bank Holiday	May 2
Bank Holiday	May 30
Bank Holiday	August 1
Christmas Day (Sun)	December 25
Boxing Day	December 26
Bank Holiday	December 27

Northern Ireland

New Year	January 3
St. Patrick's Day	March 17
Good Friday	April 1
Easter Monday	April 4
Bank Holiday	May 2
Bank Holiday	May 30
Bank Holiday	July 12
Bank Holiday	August 29
Christmas Day (Sun)	December 25
Boxing Day	December 26
Bank Holiday	December 27

Republic of Ireland

New Year	January 3
St. Patrick's Day	March 17
Good Friday	April 1
Easter Monday	April 4
Bank Holiday	June 6
Bank Holiday	August 1
Bank Holiday	October 31
Christmas Day (Sun)	December 25
St. Stephen's Day	December 26
Bank Holiday	December 27

1995 Bank & Public Holidays

May

Wk	M	T	W	T	F	S	S
18	1	2	3	4	5	6	7
19	8	9	10	11	12	13	14
20	15	16	17	18	19	20	21
21	22	23	24	25	26	27	28
22	29	30	31				

June

Wk	M	T	W	T	F	S	S
22				1	2	3	4
23	5	6	7	8	9	10	11
24	12	13	14	15	16	17	18
25	19	20	21	22	23	24	25
26	26	27	28	29	30		

November

Wk	M	T	W	T	F	S	S
44			1	2	3	4	5
45	6	7	8	9	10	11	12
46	13	14	15	16	17	18	19
47	20	21	22	23	24	25	26
48	27	28	29	30			

December

Wk	M	T	W	T	F	S	S
48					1	2	3
49	4	5	6	7	8	9	10
50	11	12	13	14	15	16	17
51	18	19	20	21	22	23	24
52	25	26	27	28	29	30	31

England and Wales

New Year	January 2
Good Friday	April 14
Easter Monday	April 17
Bank Holiday	May 1
Bank Holiday	May 29
Bank Holiday	August 28
Christmas Day	December 25
Boxing Day	December 26

Scotland

New Year	January 2
Bank Holiday	January 3
Good Friday	April 14
Bank Holiday	May 1
Bank Holiday	May 29
Bank Holiday	August 7
Christmas Day	December 25
Boxing Day	December 26

Northern Ireland

New Year	January 2
St. Patrick's Day	March 17
Good Friday	April 14
Easter Monday	April 17
Bank Holiday	May 1
Bank Holiday	May 29
Bank Holiday	July 12
Bank Holiday	August 28
Christmas Day	December 25
Boxing Day	December 26

Republic of Ireland

New Year	January 2
St. Patrick's Day	March 17
Good Friday	April 14
Easter Monday	April 17
Bank Holiday	June 5
Bank Holiday	August 7
Bank Holiday	October 30
Christmas Day	December 25
St. Stephen's Day	December 26

Metric units

Area

100 sq millimetres	= 1 sq centimetre
100 sq centimetres	= 1 sq decimetre
100 sq decimetres	= 1 sq metre
100 sq metres	= 1 are
100 ares	= 1 hectare
100 hectares	= 1 sq kilometre

Beer, wines and spirits

Proof spirit contains 57.03% pure alcohol by volume (at 50°F)

Proof strength in degrees = % of alcohol by volume (at 50°F) multiplied by 1.7535.

Beer	
nip	= ¼ pint
small	= ½ pint
large	= 1 pint
flagon	= 1 quart
anker	= 10 gallons
tun	= 216 gallons
Wines and spirits	
tot (whisky)	= ⅛, ⅙, ¼ or ⅓ gill
noggin	= 1 gill
bottle	= 1⅓ pints
Champagne	
2 bottles	= 1 magnum
4 bottles	= 1 jeroboam
20 bottles	= 1 nebuchadnezzar

Book sizes

Crown Quarto	= 246 × 189 mm
Crown Octavo	= 186 × 123 mm
Demy Quarto	= 276 × 219 mm
Demy Octavo	= 216 × 138 mm
Royal Quarto	= 312 × 237 mm
Royal Octavo	= 234 × 156 mm
A4	= 297 × 210 mm
A5	= 210 × 148 mm

Capacity

10 millilitres	= 1 centilitre
10 centilitres	= 1 decilitre
10 decilitres	= 1 litre
1 litre	= 1 cu decilitre
10 litres	= 1 dekalitre
10 dekalitres	= 1 hectolitre
10 hectolitres	= 1 kilolitre
1 kilolitre	= 1 cu metre

Energy

1000 British thermal units (Btu.)	= 0.293 kWh
100000 Btu	= 1 therm
1 UK horsepower	= 0.7457 killowatt

Length

1000 nanometres	= 1 micrometre
1000 micrometres	= 1 millimetre
10 millimetres	= 1 centimetre
10 centimetres	= 1 decimetre
1000 millimetres	= 1 metre
100 centimetres	= 1 metre
10 decimetres	= 1 metre
10 metres	= 1 dekametre
10 dekametres	= 1 hectometre
10 hectometres	= 1 kilometre
1000 kilometres	= 1 megametre
Nautical	
1852 metres	= 1 int. nautical mile

Paper sizes

Large post	= 16½ × 21 in 419.1 × 533.4 mm
Demy	= 17½ × 22½ in 444.5 × 571.5 mm
Medium	= 18 × 23 in 457.2 × 584.2 mm
Royal	= 20 × 25 in 508 × 635 mm
Double crown	= 20 × 30 in 508 × 762 mm
'A' Series (metric sizes)	
A0	= 841 × 1189 mm 33⅛ × 46¾ in
A1	= 594 × 841 mm 23⅜ × 33⅛ in
A2	= 420 × 594 mm 16½ × 23⅜ in
A3	= 297 × 420 mm 11¾ × 16½ in
A4	= 210 × 297 mm 8¼ × 11¾ in
A5	= 148 × 210 mm 5⅞ × 8¼ in

Petroleum

1 barrel	= 42 US gallons
	= 34.97 UK gallons
	= 0.159 cubic metre

Precious metals

24 carat implies pure metal.

1 metric carat	= 200 milligrams
1 fine ounce	= 480 grains
1 troy ounce	= 31.1035 grams

Volume

1000 cu millimetres	= 1 cu centimetre
1000 cu centimetres	= 1 cu decimetre
1000 cu decimetres	= 1 cu metre
1000 cu metres	= 1 cu dekametre

Weight (mass)

1000 milligrams	= 1 gram
10 grams	= 1 dekagram
10 dekagrams	= 1 hectogram
10 hectograms	= 1 kilogram
100 kilograms	= 1 quintal
1000 kilograms	= 1 tonne

Temperature Conversion

C = 5/9 (F – 32); F = 9/5C + 32.

Celsius	–18°	–10	0	10	20	30	40			
Fahrenheit	0°	10	20	32	40	50	60	70	80	90	100

Conversion Formulae

To Convert	Multiply by
Inches to Centimetres	2.540
Centimetres to Inches	0.393701
Feet to Metres	0.3048
Metres to Feet	3.2808
Yards to Metres	0.9144
Metres to Yards	1.09361
Miles to Kilometres	1.60934
Kilometres to Miles	0.621371
Sq Inches to Sq C/metres	6.4516
Sq C/metres to Sq Inches	0.155
Sq Metres to Sq Feet	10.7639
Sq Feet to Sq Metres	0.092903
Sq Yards to Sq Metres	0.836127
Sq Metres to Sq Yards	1.19599
Sq Miles to Sq Kilometres	2.58999
Sq Kilometres to Sq Miles	0.386103
Acres to Hectares	0.404678
Hectares to Acres	2.47101
Cub Inches to Cub C/metres	16.3871
Cub C/metres to Cub In	0.0610237
Cub Feet to Cub Metres	0.0283168
Cub Metres to Cub Feet	35.3147
Cub Yards to Cub Metres	0.764555
Cub Metres to Cub Yards	1.30795
Cub Inches to Litres	0.016387
Litres to Cub Inches	61.024
Gallons to Litres	4.546
Litres to Gallons	0.22
Grains to Grams	0.0648
Grams to Grains	15.43
Ounces to Grams	28.3495
Grams to Ounces	0.035274
Pounds to Grams	453.592
Grams to Pounds	0.00220462
Pounds to Kilograms	0.4536
Kilograms to Pounds	2.20462
Tons to Kilograms	1016.05
Kilograms to Tons	0.0009842

The Astronomical information in this Diary is reproduced with permission from data supplied by the Science & Engineering Council.

The Numbering of weeks is in accordance with British Standard 4760 1971

Country Code Abbreviation.

UK United Kingdom (inc. Scotland)
SCOT Scotland IRL Rep. of Ireland
CDN Canada NI Northern Ireland
USA United States of America

General Information

Phases of The Moon 1994

New Moon	d	h	m	First Quarter	d	h	m	Full Moon	d	h	m	Last Quarter	d	h	m
Jan	11	23	10	Jan	19	20	27	Jan	27	13	23	Jan	5	00	00
Feb	10	14	30	Feb	18	17	47	Feb	26	01	15	Feb	3	08	06
Mar	12	07	05	Mar	20	12	14	Mar	27	11	09	Mar	4	16	53
Apr	11	00	17	Apr	19	02	34	Apr	25	19	45	Apr	3	02	55
May	10	17	07	May	18	12	50	May	25	03	39	May	2	14	32
June	9	08	26	June	16	19	56	June	23	11	33	June	1	04	02
July	8	21	37	July	16	01	12	July	22	20	16	June	30	19	31
Aug	7	08	45	Aug	14	05	57	Aug	21	06	47	July	30	12	40
Sept	5	18	33	Sept	12	11	34	Sept	19	20	00	Aug	29	06	41
Oct	5	03	55	Oct	11	19	17	Oct	19	12	18	Sept	28	00	23
Nov	3	13	35	Nov	10	06	14	Nov	18	06	57	Oct	27	16	44
Dec	2	23	54	Dec	9	21	06	Dec	18	02	17	Nov	26	07	04
												Dec	25	19	06

All times shown are G.M.T. Add 1 hour between March 27th and October 23rd to allow for BST.

Time Differences

Hours plus or minus GMT in London

Accra	00.00	Colombo	+05.30	Lisbon	00.00	San Francisco	−08.00
Adelaide	+09.30	Copenhagen	+01.00	Los Angeles	−08.00	Santiago	−04.00
Alexandria	+02.00	Delhi	+05.30	Madrid	+01.00	Shanghai	+08.00
Amsterdam	+01.00	Detroit	−05.00	Malta	+01.00	Sierra Leone	00.00
Athens	+02.00	Durban	+02.00	Mauritius	+04.00	Singapore	+08.00
Baghdad	+03.00	Gibraltar	+01.00	Melbourne	+10.00	Stockholm	+01.00
Bangkok	+07.00	Halifax	−04.00	Montevideo	−03.00	St. Petersburg	+03.00
Beijing (Peking)	+8.00	Helsinki	+02.00	Montreal	−05.00	Sydney N.S.W.	+10.00
Bombay	+05.30	Hong Kong	+08.00	Moscow	+03.00	Tehran	+03.30
Buenos Aires	−03.00	Honolulu	−10.00	Nairobi	+03.00	Tokyo	+09.00
Cairo	+02.00	Houston	−06.00	New York	−05.00	Toronto	−05.00
Calcutta	+05.30	Istanbul	+02.00	Oslo	+01.00	Vancouver	−08.00
Calgary	−07.00	Jakarta	+07.00	Perth W.A.	+08.00	Wellington	+12.00
Cape Town	+02.00	Karachi	+05.00	Rangoon	+06.30	Winnipeg	−06.00
Chicago	−06.00	Lagos	+01.00	Rio de Janeiro	−03.00	Yokohama	+09.00
Christchurch N.Z.	+12.00	Lima	−05.00	Rome	+01.00	Zurich	+01.00

The times listed above compare the standard (winter) times in the various cities. Some countries adopt Summer (Daylight Saving) Time i.e. +1 hour, for part of the year.
British Summer Time (1 hour in advance of GMT) will be observed in the U.K. between 01.00 hrs on March 27th and 01.00 hrs on October 23rd.

International Telephone Codes

Country	Country Code Dialling To	Access Code Dialling Out
Australia	61	0011
Austria	43	00
Bahrain	973	00
Belgium	32	00
Brazil	55	00
Canada	1	011
China	86	00
Czech Rep.	42	00
Denmark	45	009
Egypt	20	00
Finland	358	990
France	33	19
Germany	49	00
Greece	30	00
Hong Kong	852	001
Hungary	36	00
India	91	900
Iran	98	00
Iraq	964	00
Israel	972	00
Italy	39	00
Japan	81	001
Kuwait	965	00
Luxembourg	352	00
Malta	356	0
Mexico	52	00
Monaco	3393	19
Morocco	212	00
Netherlands	31	09
New Zealand	64	00
Nigeria	234	009
Norway	47	095
Poland	48	00
Portugal	351	07
Russian Federaton	7	810
Saudi Arabia	966	00
Singapore	65	005
Slovak Rep.	38	00
South Africa	27	091
Spain	34	07
Sweden	46	009
Switzerland	41	00
Taiwan	886	002
Trinidad & Tobago	1809	01
Tunisia	216	00
Turkey	90	99
U.K.	44	010
U.S.A.	1	011
Venezuela	58	00

Many countries are now linked to the International Direct Dialling System. To dial direct use the following sequence: Access Code; Country Code; Area Code; Number.

Example	Access Code	Country Code	Area Code	Number
UK to Naples	010	39	(0)81	.. .. ..
Italy to Birmingham	00	44	021	... .

When dialling most Countries omit the prefix digit '0' from the Area Code

Cash Account · January

Debit			Credit		
WIRE	CCTV		(HOLE)		
GRN/WHT	→		GRN/WHT		
GRN	→		GRN		
BRN/WHT	→		BLU/WHT		
BLU/WHT	→		ORN		
BLU	→		BLU		
ORN	→		ORN/WHT		
WIRE	PORT		TEST		
GREEN	—		INVERTER		
ORN	—		INVERTED		
ORN/WHT	=		-2		
BLUE	=		-3		
BLU/WHT	=		-1		
Total			Total		

Cash Account · February

System Mappings

Debit		Credit
G	FRED STORE	— ~~USER~~ AGENT = REX
H	FRED SYSTEM	
I		
J		
K		
L		
M	~~B~~ FANG APPS	— AGENT = ROVER
N	FANG ARC	
O	FANG SYSTEM	
P		
Q		
R		
S		
T	~~REDOX~~ FLAMINGO ACCESS	— AGENT = REDOX
U		
V		
W		
X		
Y		
Z		
	WORK GROUP = TCS-UK2	ID = 496583877
Total		Total

Cash Account · March

Debit				Credit			
Total				Total			

Cash Account · April

(VIDEO CAPTURE UNIT)

Debit				Credit			
	BS BG						
279	= LS, FS, LUM (HIGH)						
37A	= [redacted]. HOR POS (MSB) [0 = Low field, 1 = High field]						
288	= LINE No.						
379	= LUM (?)						
27A	= READ / WRITE (0 = WRITE, 1 = READ)						
378	= HOR. POS.						
	<(GREEN PCB)>						
Total				Total			

Cash Account · May

Beginning

Debit			Credit		
	79, 95, 127				
	1001111	-79			
	1011111	-95			
	1111111	-127			
	1001111	— 79			
	1011111	— 95			
	1111111	— 127			
	FS/LS =	MASK =		48d	30H
279H	Bit a =	32d		20H	- (LS?
	Bit b =	47d		2FH	- (FS?
279H	Bit 0 =	FRAME No.			
				1001111	
				11001111	-207 (FS)
				1011111	-95 (LS)
				LS = 10H ✓	
				FS = 80H ✓	32D
Total			Total		

7th July

FD

with logo

Mary Bea

DONE.

Cash Account · June

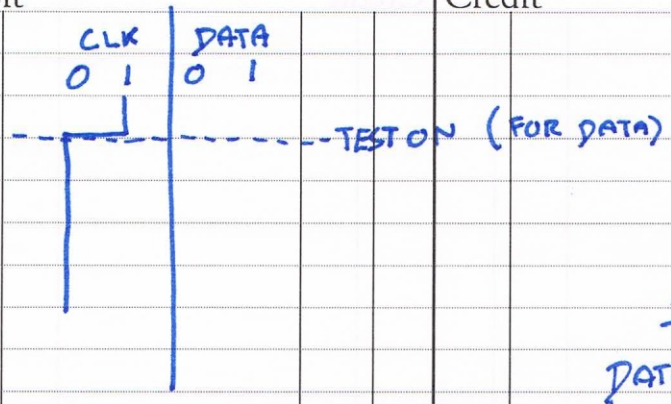
Debit				Credit			
Total				Total			

Cash Account · July

Debit				Credit			
Total				Total			

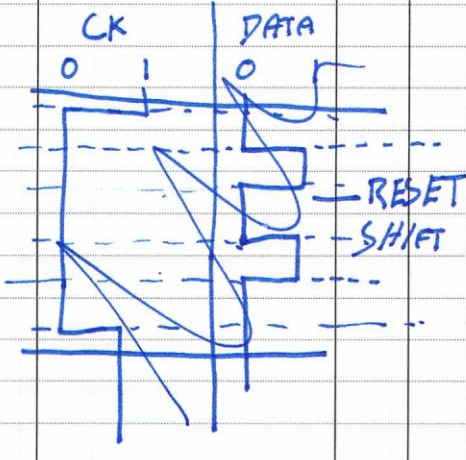
Cash Account · August

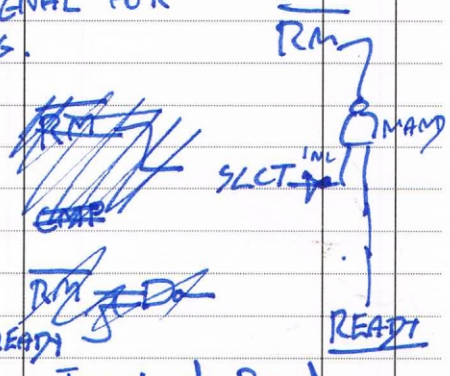
Debit	Credit



SLCT = 0
= RESET

DATA can only enter for Shift Register when SLCT = 0





RESET ON:
SLCT = 1 AND READY = 0

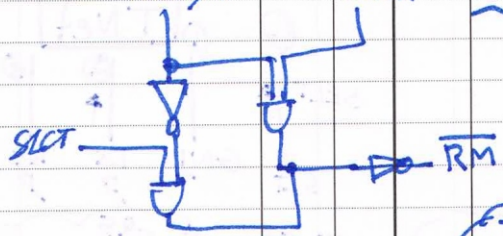
(SLCT = 1 AND CLK = 1
READY = 1) = End of Terminal Ready

(SLCT = 1 AND READY = 0) = START OF NEW PACKET

Total	Total

Cash Account · September

Debit				Credit			



AND/INV CIRCUIT

TO RESET

SERVER SENDS A PULSE ON SLCT, TO END COMMS SERVER SENDS A CLK PULSE.

TO RESET

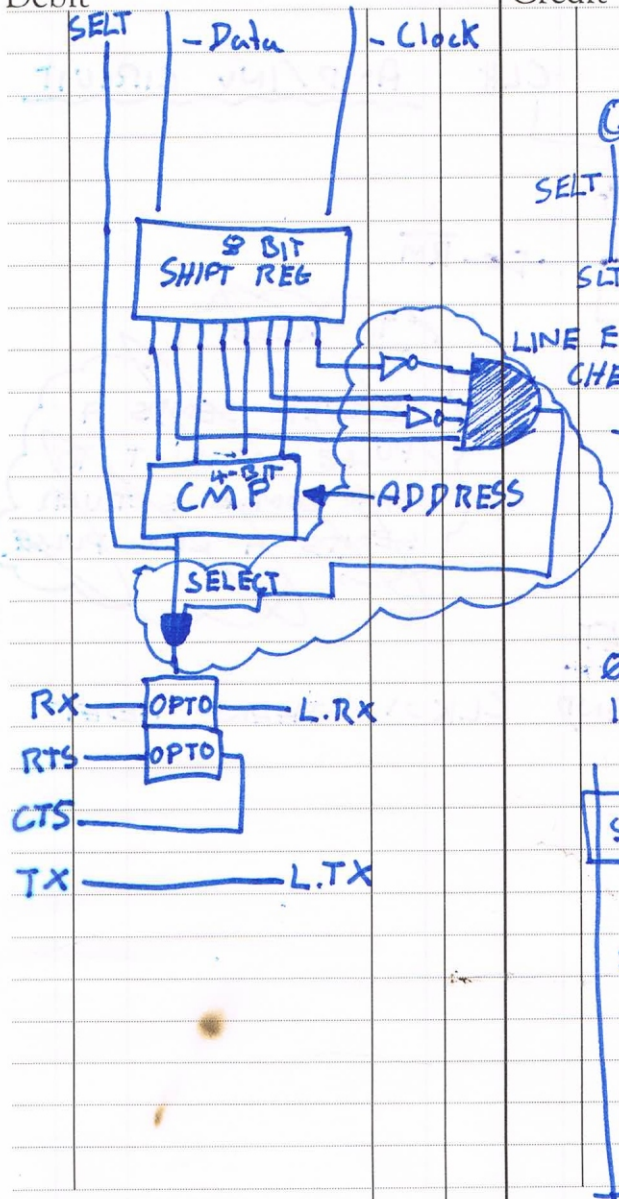
SLCT =1 AND CLK=1 THEN RESET!

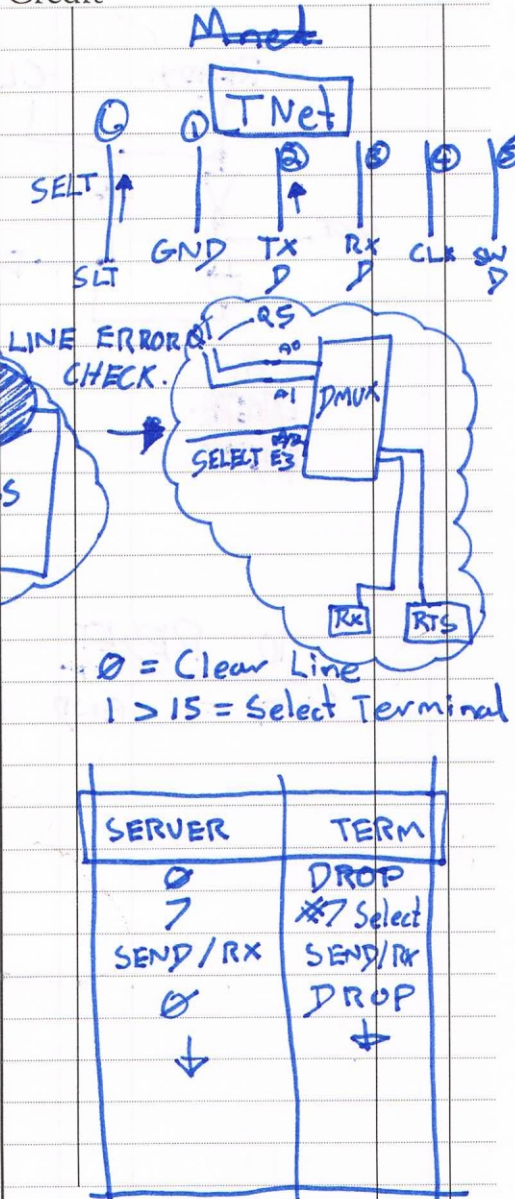
Total				Total			

Cash Account · October

Packet Switching Network

Debit | Credit



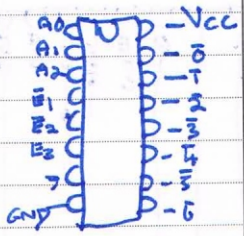


Total | Total

Chips

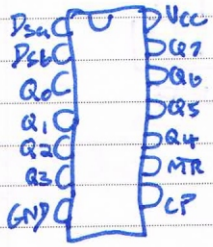
Debit		Credit

SN74LS138N (3 → 8 De-Mux)



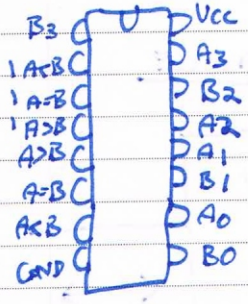
74LS30N
(8 INPUT NAND)
7 = OUT
14 = Vcc
7 = GND

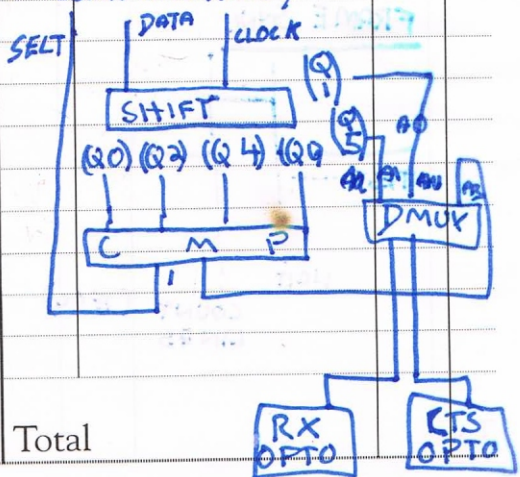
SN74LS164N (8 Bit Shift Reg)



74LS04N
(HEX INVERTER)
7 GND 1-2 9-8
14 VCC 3-4
5-6
13-12
11-10

SN74LS85N (4BIT COMPARATOR)



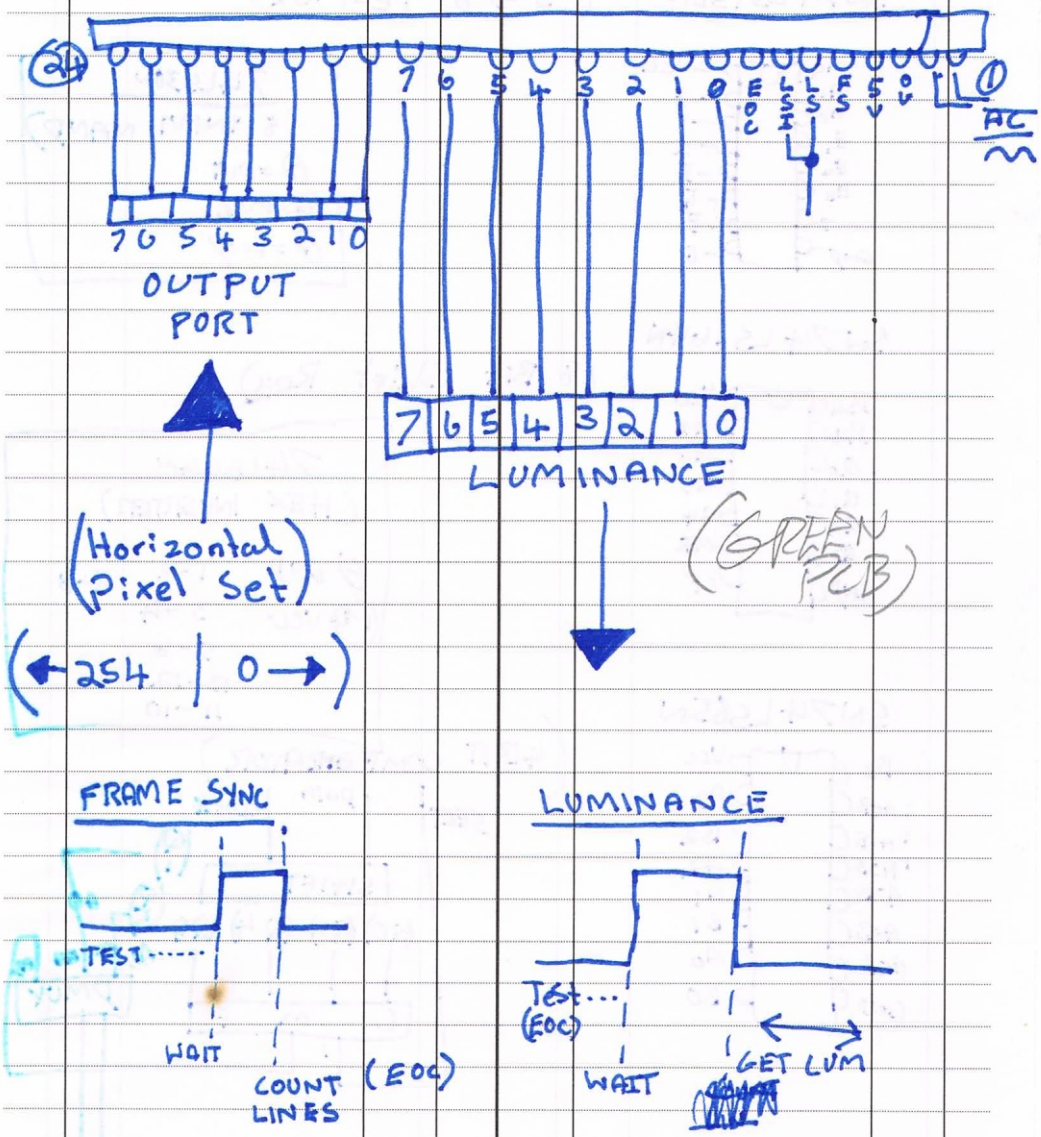


Total | Total

Cash Account · December

Digitiser (MAX 3000 – 255 = 255 B/W)

Debit			Credit			



Total			Total			

Annual Cash Summary

(* DIGITISER PORT MAP *)

Balance Brought Forward				
	(LPT)			
January	PORT D_0, D_7	= Horizontal Pixel Set		
February	(10) Acknowledge	= Luminance (0)		
	(11) Busy	= Luminance (1)		
March	(12) Paper out	= Luminance (2)		
	(13) Select	= Luminance (3)		
April	(15) Error / 14*	= Luminance (4)		
May				
June	(COM)			
July	Clear To Send	= LS - LINE SYNC / LS1		
	Data Set Ready	= FS - FRAME SYNC		
August	CARRIER DETECT	= EOC		
September				
	FS = 15*			
October	LS/LS1 = 16*			
	EOC = 17*			
November				
December				
Total				
Balance Carried Forward				

Statistical Summary

QUME

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
27	91			HOME								
27	99			?								

Notes

TO CPU

LUM	LEV
GRN	DARKEST (-4, -5 INT)
GRN/WHT	DARKEST
ORANGE	LIGHT (0 INT)
WHT/BLUE	DIM (-1 INT)
WHT/ORN	DIM (-2 INT)
BLUE	DIM (-3 INT)

FROM CCTV

LUM	LEV
GRN	
GRN/WHT	
ORANGE	MID
WHT/BLUE	LUM -2 ✓
WHT/ORN	LUM -1
BLACK	MID

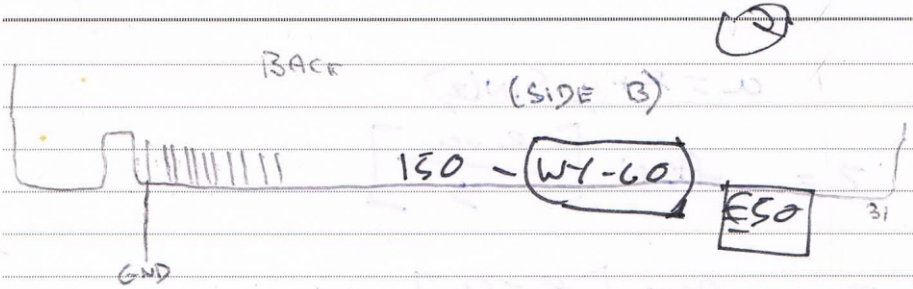
$$a = e + y\,\mathrm{SIN}(z)$$

$$z = -\mathrm{ASIN}\left[\frac{e-a}{y}\right]$$

$$z = -\mathrm{ASIN}\left[\frac{e-a}{y}\right] - \pi$$

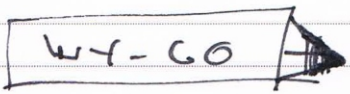
$$z = -\mathrm{ASIN}\left[\frac{e-a}{y}\right] + \pi.$$

Notes PC CARD PIN OUT



1 GND	16
2 RESET	17
3 +5V	18
4 IRQ2	19
5 -5V	20
6	21 IRQ7
7 -12V	22
8	23 IRQ5
9 +12V	24 IRQ4
10 GND	25 IRQ3
11	26
12	27
13 $\overline{TOW}$	28
14 $\overline{TOR}$	29 +5V
15	30 BCLOCK 14.318180MHZ INP
	31 GND

5-6 years



Amber Screen

Notes

PC CARD PIN OUT

SIDE A (COMPONENT SIDE)

1		16	A15
2	D0	17	A14
3	D1	18	A13
4	D2	19	A12
5	D3	20	A11
6	D4	21	A10
7	D5	22	A9
8	D6	23	A8
9	D7	24	A7
10		25	A6
11	AEN	26	A5
12	A19	27	A4
13	A18	28	A3
14	A17	29	A2
15	A16	30	A1
		31	A0

~~MAR DC-UOUTT~~

MAR DC1

Line Pos. IN

0	1	2	3	4	5	6	7
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0

Addresses & Telephone Numbers

ftp. :/developer/win32dk/
sdk public/Win32s115a.zip

ftp. i/developer/drg/WinG/
WinG10.ZIP

ftp

.46.81.37916

ftp://ftp. .com/dirmap.htm

232.1 3081

Dec/Jan 1993/94

WEEK 52

27 Monday

DAY (361-4)
Boxing Day Holiday [UK] [IRL] [CDN]

28 Tuesday

DAY (362-3)
Holiday [UK] [IRL] [CDN]

29 Wednesday

DAY (363-2)

30 Thursday

DAY (364-1)

31 Friday

DAY (365)

1 Saturday

DAY (1-364)
New Year's Day

2 Sunday

DAY (2-363)
2nd Sunday after Christmas

Memorandum

Willkommen zu die Inet, es das ein neu interface zwischen die Benutzer und die coputer.

Da sind viele auf die Markt.

WEEK 1

The True Environment File.

3 Monday
DAY (3-362)
Holiday [UK] [IRL] [USA] [CDN]

The True Environment File contains all the information required to start up the Command Line Shell. This includes:

4 Tuesday
DAY (4-361)
Holiday [SCOT]

- COLS
- LINES
- TERMINAL INITIALISATION STRING
- PWD
- → (Expandable)

5 Wednesday
DAY (5-360)

The Filename of this file is .Tsh.Dat
A Hidden File. (can not be edited in the normal ways).

(Dalby) is due 12/1.

6 Thursday
DAY (6-359)
Epiphany

User Startup File

This file (if present) is executed before the shell starts proper. Any Aliases, Variables etc. get by this startup program can not be altered afterwards.

7 Friday
DAY (7-358)

The file is just a normal program but the special command
.'STARTUP (filename) — must be used to specify a new startup file.

8 Saturday
DAY (8-357)

To startup a 'clean shell' you may press the following keys

9 Sunday
DAY (9-356)
1st Sunday after Epiphany

[CNTRL] + [N] + [CNTRL] + [O] + [F1]

TRUE COMMAND LINE SYSTEM

Commands are either internal or external. External commands are all stored in a set path which also contains a master file which matches the commands to a 'parse code'. Alternatively each command has its own 'parse code' at the last two bytes of its COM file. Eg.

[COM][ ] — (code, sub-code)

When writing a program the editor looks in the set path for external commands. If a command exists, the editor retrieves the (code and sub-code) and uses these in the source code.

External commands must have sub-codes between (10 and 128). It may have any (code).

If an external does not exist then the editor assumes the keyword is an alias. The priority list is as follows:

(1) Internal Commands
(2) External Commands
(3) Aliases.

January 1994

WEEK 2

Memory Image External Commands

10 Monday
DAY (10-355)

Commands must conform to the following standards:

11 Tuesday
DAY (11-354)

- ALL Input and Output is via the 'transfer area' and

- No input can be assumed except from the original transfer input area, when the command has first executed.

12 Wednesday
DAY (12-353)

- No output to screen, Lpt or any device. (unless this is the commands ONLY role. ie, a Lpt command to send data to the printer spooler.

13 Thursday
DAY (13-352)

- Commands may not call other commands.

14 Friday
DAY (14-351)

- Commands must assume their start address to be SEG:0000

15 Saturday
DAY (15-350)

- Commands should be less than 64k in length

16 Sunday
DAY (16-349)
2nd Sunday after Epiphany

- When accessing disk/Lpt or other devices, information on the device must be given in the input.

External Commands For True

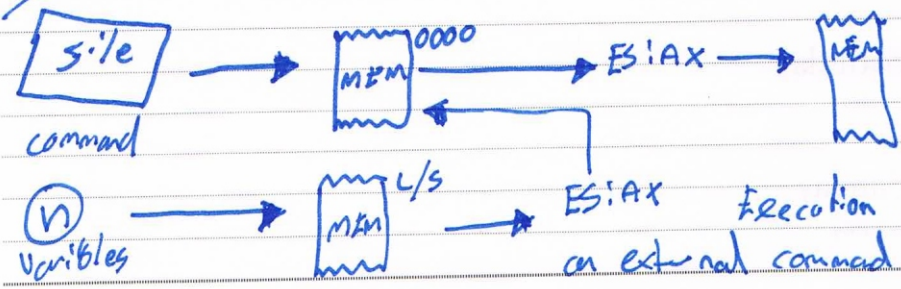
True uses files to pass command line information to the external command. A standard file in the 'External Command Path' is always used. This file contains:

- Process ID (for internal use)
- Varibles Decoded for Input.
- Varible codes for output
- STDOUT Device for Command
- STDIN Device for Command

Memory Image Commands

These commands are machine code files which are loaded into memory during execution. Data is passed to these commands by setting the ES:AX to the segment where this 'image' data exists.

Output from the command is also placed at a position within this area. (STDOUT) STDIN must be first gathered from the getch ~~comm~~ internal command.



Execution of an external command.

WEEK 3

Memory image Commands

(continued..)

17 Monday
DAY (17-348)
Martin Luther King Jr. Birthday
Observance, Holiday [USA]

o Commands must be able to exit successfully, no matter what input they receive.

18 Tuesday
DAY (18-347)

o All information required by a command must be supplied upon its execution.

19 Wednesday
DAY (19-346)

20 Thursday
DAY (20-345)

21 Friday
DAY (21-344)

22 Saturday
DAY (22-343)

23 Sunday
DAY (23-342)
3rd Sunday after Epiphany

Memorandum

Memory image Commands

When a command executes, it may require 'system' information on such items as the current drive, current printer, terminals ID etc.. A range of such information is made available to all commands at the first 2K of the transient area. Amongst other items this area contains:

- o Current working directory (PWD)
- o Random Number Integer
- o Date
- o MPATH$, USERPATH$, DRIVE
- o SPOOLER PATH. FILENAME$
- o COM PORT for terminal, + setup string
- o COLS, LINES, USER$, USERID$, PASSWD$
- o Login file, Permissions

NB: These varibles may be set by the command, and when execution is complete, the varibles are read back into the system.

January 1994

WEEK 4

24 Monday
DAY (24-341)

25 Tuesday
DAY (25-340)

26 Wednesday
DAY (26-339)

27 Thursday
DAY (27-338)

28 Friday
DAY (28-337)

29 Saturday
DAY (29-336)

30 Sunday
DAY (30-335)
9th Sunday before Easter

Run List . DAT

This file contains the names of all installed external commands and the allocated functions and sub-functions for them.

eg.

add. 17. 1 [CRLF]

Case sensitive

Separator

Space

Func No.

Each entry is in ASCII format and are separated by [CR] (13).

The editor 'Tsh' uses this list when establishing which externals to use.

Calling/writing Memory Image Commands

31 Monday
DAY (31-334)

Memory Image Commands are structured as internally defined interrupts. They are allocated on System interrupts, and are executed from this. Therefore they must end with an IRET.

1 Tuesday
DAY (32-333)

CALL! AX = POSITION START OF TRANSFER AREA

BX = POSITITION OF END OF INPUT VERIBLS.

DS = SEGMENT OF TRANSFER AREA.

2 Wednesday
DAY (33-332)

~~PROGR~~ COMMAND IMAGE FILE HEADER

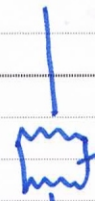
where ~~OTHA~~ command can start its output varibles.

0000 – 0002 JMP
0003 Sub func (>10, <128)
0004 func
0005 No. of Input varibles
0006 No. of Output varibles

optional

3 Thursday
DAY (34-331)

4 Friday
DAY (35-330)



Input varibles (Inorder)
(1), (2), (3) – See next page.

Output varibles
(1), (2), (3) – see next page

5 Saturday
DAY (36-329)

PROGRAM STARTS HERE.

6 Sunday
DAY (37-328)
8th Sunday before Easter

Memorandum

Offset.

TRANSFER AREA (64K)

3000H: 1st varible

SINGLE BYTE (1)
DOUBLE BYTE (2)
STRING. (3)*
~~HALF BYTE (4)~~

(4) = Memory Image
{2 Bytes} segment {2 Bytes} offset

Output varible information begins at the end of the input varible list. At the end of the last varible for input.

* For strings - Command files use NULL terminated strings of unspecified length.

The sync, sync in the header is used during runtime to check the validity of the Run List. The 'Run List Compiler' is a utility which sets these ~~areas~~ varibles in all the command files, and maintains a 'Run List' which the Tsh uses when giving an external command.

The Run List compiler will not change the ~~sync~~ commands header unless two identical headers are present.

A standard OBJECT list may be use to set all commands to a set standard. without this list, programs written under Tsh may be un compatible when using external commands.

February 1994

©1993-94

WEEK 6

Tnet operations for PSTN

7 Monday
DAY (38-327)

All data sent is processed according to the clients 'mode'. Two modes exist, 'Real' mode and 'command' mode.

8 Tuesday
DAY (39-326)

1) BASIC STRUCTURE

Command mode

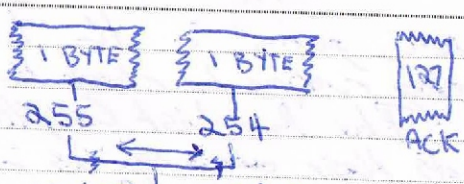
The (255) or 'Escape' character is used to bring the system to command level.

9 Wednesday
DAY (40-325)

A return Byte of (127) is sent from client to acknowledge the mode changing to 'command mode'.

10 Thursday
DAY (41-324)

Real mode



>0.25sec, <0.8sec

11 Friday
DAY (42-323)

12 Saturday
DAY (43-322)
Lincoln's Birthday (1809) [USA]
Ramadan, First day of

NB: COMMAND MODE CHR (255).

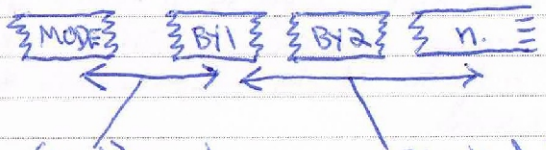
when ever this character is sent the client drops any previous process and responds with a new [ACK] code.

13 Sunday
DAY (44-321)
7th Sunday before Easter

Memorandum

Timing

Where data being sent is in string form, each byte must ~~mak~~ conform to the following timing for data.

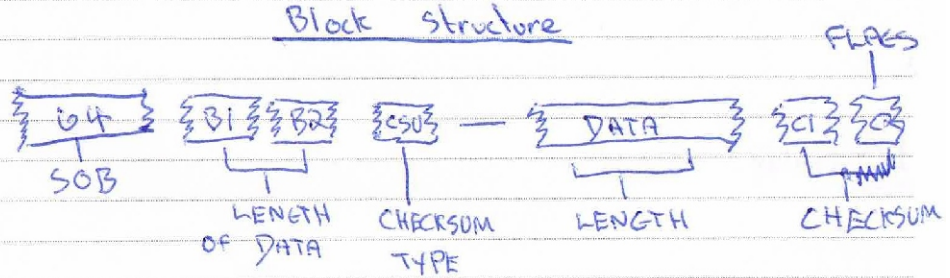


System (Host) must timeout after a period of 1sec if no [ACK] is sent.

Block data must be sent with no more than 0.25 secs. between bytes.

A standard timeout of '0.25' secs. is in operation on all bytes except where acknowledgment is required, where a timeout of 1sec is used.

Block structure



NB: NO (255) CHARACTER CAN BE SENT. FOR CHARACTERS OUTSIDE OF THE ~~DATA~~ THE [FLAGS] BYTE CONTAINS THE FLAGS.

7	6	5	4	3	2	1	0
					LN1	LN2	CSM

LN1, LN2, CSM

WHEN BIT IS '1' 1 IS ADDED TO THE DATA BYTE.

February 1994 XRE FILES (255 COLOUR)

WEEK 7

BLOCK STRUCTURE

14 Monday
DAY (45-320)
St. Valentine's Day

LENGTH

MSB B1 LSB B2

Total block length is governed by the total values of B1 + B2, Max. data length is therefore 65000 bytes.

15 Tuesday
DAY (46-319)
Shrove Tuesday

However because of PSTN transmission methods blocks of more than 8000 bytes should not be used.

16 Wednesday
DAY (47-318)
Ash Wednesday

LATE AFRT.

17 Thursday
DAY (48-317)

18 Friday
DAY (49-316)

19 Saturday
DAY (50-315)

20 Sunday
DAY (51-314)
1st Sunday in Lent

Memorandum

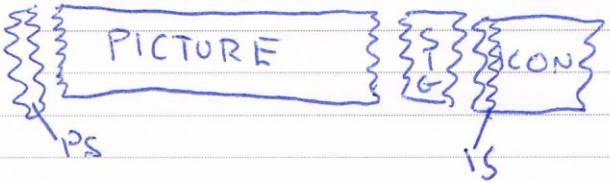
February 1994

XRE FILE FORMAT

4/3 A.R.

WEEK 8

21 Monday
DAY (52-313)
Presidents' Day, Holiday [USA]



22 Tuesday
DAY (53-312)
Washington's Birthday (1732)

EXTESION = XRE.

23 Wednesday
DAY (54-311)

24 Thursday
DAY (55-310)

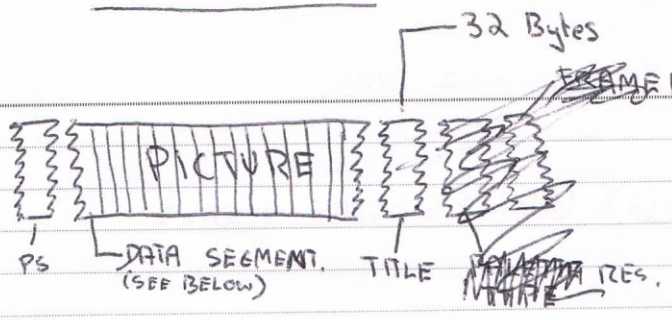
25 Friday
DAY (56-309)
Purim

26 Saturday
DAY (57-308)

27 Sunday
DAY (58-307)
2nd Sunday in Lent

16/9 FORMAT

Memorandum



0 DATA SEGMENT

(0) PALETTE No. (1) = 255 XRE (2) = C+Y (3) = 255 XRE2.

(1) No. of Lines (200, 400, 760, 1024, 2048)

(2) No. of Pixels (320, 640, 711, 1351, 1820, 3640)

(3) Frame No. (Only for Motion)

(4) Aspect Ratio (1) = 4/3 (2) = 16/9

(5) Sound Link

(6) Interaction Link

(7) Pause time in ms.

(8) R.

(9) R.

(10) → Frame image

Feb/March 1994

WEEK 9

VIDEO FUSION MK2 XRE

28 Monday
DAY (59-306)

BUF(1), BUF(2), BUF(3) = FRAME STORES

1 Tuesday
DAY (60-305)
St. David's Day

[1] DISPLAY BITMAP
+ [BUF], + [X], + [Y], [EXP]

0 = OVERLAY
EXP = 1 = PSET, 2 OR, 3 XOR

2 Wednesday
DAY (61-304)

[2] DISPLAY RAW TEXT
+ [BUF], + [X], + [Y], [COLOUR], [FONT]

3 Thursday
DAY (62-303)

[3] LOAD BUFFER
+ [BUF], + [FILENAME]

4 Friday
DAY (63-302)

[4] MOVE SCREEN AREA
[XS], [YS], [XE], [YE], [NXP], [NYP], [EXP]

5 Saturday
DAY (64-301)

LOOP:
COUNT = NUMBER OF EXECUTIONS OF THE LOOP
LOOPS = LINE NUMBER OF START OF LOOP.
LOOPE = LINE NUMBER OF END OF LOOP.

6 Sunday
DAY (65-300)
3rd Sunday in Lent

AFTER LOOP IS COMPLETED, EXECUTION CONTINUES AFTER THE 'LOOP' COMMAND.

[5] GET SCREEN
[XS], [YS], [XE], [YE], [BUF]

[6] GET BUFFER
[BUF], ~~[YS], [XE],~~ [XS], [YS], [XE], [YE], [BUF]
SOURCE TARGET

[7] FLIP BUFFER
[BUF], [FLAG] FLAG = 1, L⇔R / 2 = H⇔V

[8] REDUCE BUFFER
[BUF], [XC], [YC], [Z] 0/1 = NORMAL, 255 = MAX

[9] ENLARGE BUFFER
[BUF], [XC], [YC], [Z] 0/1 = NORMAL, 255 = MAX

[10] FILTER COLOURS
[BUF], [XS], [YS], [XE], [YE], [FILTER]

[11] BOX
[BUF], [XS], [YS], [XE], [YE], [COLOUR], [FILL], [FILL COLOUR]

[12] CIRCLE / ELIPSE
[BUF], [X], [Y], [YSZ], [XSZ], [COLOUR], [FILL], [FILL COLOUR]

[13] SPEED
[SPEED CONSTANT]

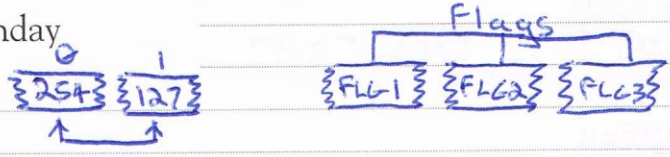
[14] LOOP
[COUNT] [GOTO LOOPS] [GOTO LOOPE]

March 1994

TRUX.

WEEK 10 TRUX File Permissions

7 Monday
DAY (66-299)

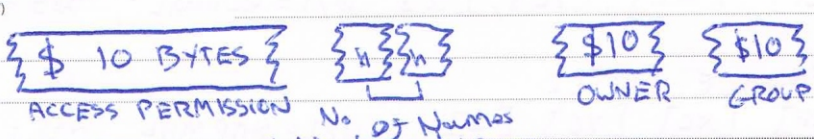


Two Bytes indicate this is a Trux file.

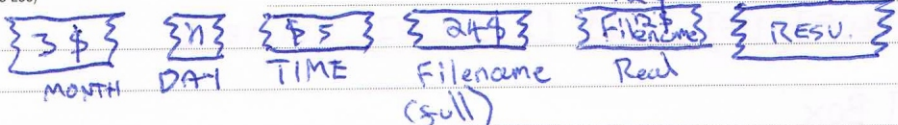
8 Tuesday
DAY (67-298)

Notes: Upper and Lowercase numbers are distinct.

9 Wednesday
DAY (68-297)



10 Thursday
DAY (69-296)



11 Friday
DAY (70-295)

MONTH, DAY AND TIME IN FILES

MONTH $3 ASC.II 3 CHR STRING DENOTING THE MONTH. EG Feb, Jul.

DAY n LSB = DAY 0=SUN 6=SAT
MSB = FLAGS

12 Saturday
DAY (71-294)

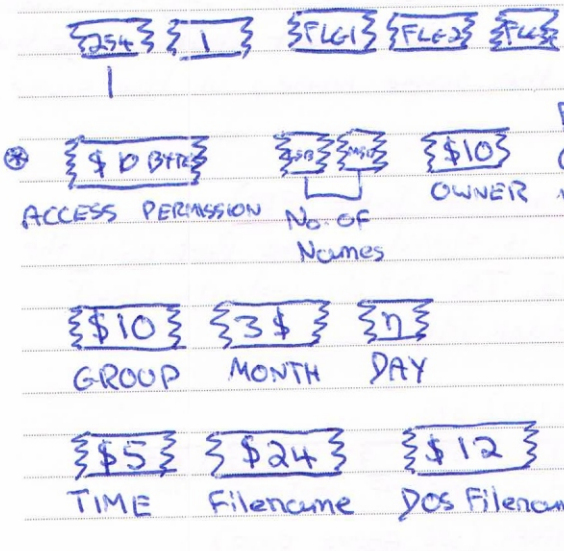
FLAGS = (WEEKLY, MON-FRI, SAT-SUN, MWF)
7 6 5 4

TIME $5 FIRST 4 BYTES = HR$ + MN$

13 Sunday
DAY (72-293)
4th Sunday in Lent
Mothering Sunday UK

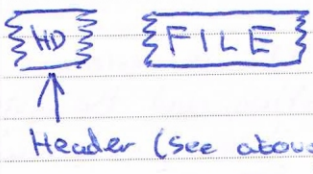
LAST BYTE = FLAGS 7
(AM/PM, SUMMER, (TZ+1), 4 → 0 TIME ZONE)
7 6 5

TRUX File structure (ADDED TO FILE HEADER OR STORED IN SPECIAL DESCRITOR FILE IN THE ROOT OF THE FILES DRIVE / PATH)



FILES CREATED BY TRUX ARE GIVEN INFORMATION IN THEIR HEADER, UNLESS THE FILE HAS A FILENAME GREATER THAN 12 CHRS (OR NON DOS FILENAME)

File:



No further search if TRUX finds the (254)(1) combination and no errors occur during header search.

No (254)(1) at start or error sends TRUX into a search for the File Discriptor Table (FDT), which is stored in the Root of this files path. If no FDT exists TRUX creates one, if FDT exists but the filesname is not matched then this is appended to the FDTs contents.

The purpose of using both a FDT and File Headers is to enable a degree of portability between users, who use TRUX, and TRUX to external files. in order to prevent both files being listed in the FDT and having their own header, a check is made by the system for the files

March 1994

WEEK 11 TRUX File Structure

14 Monday
DAY (73-292)

filenamame inside the FDT. System cant erase the file without checking ~~so~~ the FDT for other files of the same name, in the same path.

15 Tuesday
DAY (74-291)

File Discriptor Table (FDT)

The FDT is stored in the Root using the filename FDT(CHR 127).SYS. The 127 chr instructs TRUX not to show the file (Hidden File).

16 Wednesday
DAY (75-290)

FDT(CHR 127).SYS

HEADER → [254] [1] [$8] [$10] [$12]

CODE — DATE UP — LAST USER — RESERV.

⊛ ↓ FILES ENTRES (SEE ABOVE DATA)

17 Thursday
DAY (76-289)
St. Patrick's Day, Holiday [NI] [IRL]

When the FDT is referenced for a files information, and the system has details of a paths files then checking of erased files with respect of the FDT occurs. FDT entres which no longer exist, are removed from the FDT.

18 Friday
DAY (77-288)

Files which are new to the FDT and have no options specified are given the following default values:

-rw-r--r-- = {$10 ACCESS PERM}

19 Saturday
DAY (78-287)

20 Sunday
DAY (79-286)
5th Sunday in Lent

File Descriptors

TRUX

3 'handles' are open when TRUX is first started

these are:

R	0 - Standard Input	STDIN
W	1 - Standard Output	STDOUT
R/W	2 - Standard Error	STDERR

March 1994

WEEK 12

INTERNAL COMMAND STRUCTURE

21 Monday
DAY (80-285) >OUT <IN

ARGUPARSE:

OPS > Number of arguments

OPS(n) > Argument Number (see below)

OPS$(n) > Argument

22 Tuesday
DAY (81-284)

COMARGU < Number of Arg's to Search

COMARGU$< ARGUMENT CHARACTER T

COMARGUL< Length of input.

23 Wednesday
DAY (82-283)

OPS(n)= 1 - STRING

2 - COMMAND ARGUMENT

3 - OUTPUT REDIRECTION

4 - NON DESCRIP

9 - NEWCOMMAND.

24 Thursday
DAY (83-282)

8 - ASINC. COMMAND

7 - Exec. if True / NEW COM

6 - Exec. if False / NEW C

*5 - Operators

25 Friday
DAY (84-281)

* eg. output / input redirection. May be global operators sent to all commands when situated within brackets.

-1 REDIRECTED INPUT

-2 REDIRECTED OUTPUT

26 Saturday
DAY (85-280)

27 Sunday
DAY (86-279)

Palm Sunday, Pesach

British Summer Time begins, EC Summer Time begins

COMMAND PARSING (INTERNAL)

Eg. { read -r line1; read -r line2; cat; } < foobar

Parsing occurs in degenerate stages:

① Breakdown line into Braces, Setup the Variables for the Braced segments and parse those segments seperately.

eg.
BRK(1) { read -r line1; read -r line2; cat; }
BRK(2) < foobar

All BRK(1) commands share the same output destination which in this case is stdin. This is the case unless the command within the brackets/braces is directed explicitly.

② Breakdown of commands within the Braces/Brackets. (COMMANDS ARE SEPARATED, OPTIONS DECODED).

Each Breakdown is given a number which is one more than the last Breakdown / command. Execution of commands is done in this order. Any operators outside of the brackets are added to each command.

~~③ All undirected output from each command is stored in the Temporary file '.KSH(127).TEM'. If no > exists, '.KSH(127).TEM' is sent to STDOUT.~~

March/April 1994

WEEK 13

TRE-TERM 940

28 Monday
DAY (87-278)

(27) + (01) OPEN window.
+ (SY~~[illegible]~~\2) + (SX\4)
+ (LNTIT$) + (TIT$)
~~[illegible]~~

29 Tuesday
DAY (88-277)

30 Wednesday
DAY (89-276)

31 Thursday
DAY (90-275)

1 Friday
DAY (91-274)
Good Friday

2 Saturday
DAY (92-273)

3 Sunday
DAY (93-272)
Easter Day
Daylight Saving Time begins [USA] [CDN]

for loops in Tcsh

The following construct is used in Tcsh for for loops.

```
for (a>0), a+1
@)
  prn a
@)
```

increase/step

condition statement

for program brackets

program area

April 1994

WEEK 14

4 Monday

DAY (94-271)
Easter Monday, Holiday [UK]
(except Scotland) [IRL] [CDN]

5 Tuesday

DAY (95-270)

6 Wednesday

DAY (96-269)

7 Thursday

DAY (97-268)

8 Friday

DAY (98-267)

9 Saturday

DAY (99-266)

10 Sunday

DAY (100-265)
Low Sunday

Memorandum

April 1994

WEEK 15

11 Monday
DAY (101-264)

Set text cursor position

[27] [y;xf

12 Tuesday
DAY (102-263)

[27] 2K Erase line

[27] [n;nm Set text mode

13 Wednesday
DAY (103-262)

GRAPHICS

[27] [nc Set text colour (0-255) (adding another unit sets the background)

[27] [n gc Set graphics colour (0-255)

14 Thursday
DAY (104-261)

[27] [y;x gf Set graphics position

[27] [n;n gm set graphics modes

15 Friday
DAY (105-260)

- graphics modes (extended)

10 = 320x200x16 (6 pages)

12 = ~~320~~ 640x200x16 (4 pages)

12 = 640x480x16 (1 page)

13 = 320x200x256 (1 page)

14 = 640x350x16 (2 pages)

16 Saturday
DAY (106-259)

17 Sunday
DAY (107-258)
2nd Sunday after Easter

Memorandum

OR FROM CURRENT GRAPHICS POSITION

[27] E $\overbrace{y ; x}^{\times 2}$; t gl Draw line

[27] [y ; x ; s ; t gc Draw Circle / Ellipse

[27] [$\overbrace{y ; x}^{\times 2}$; t gb Draw Box

[27] [y ; x gp Draw Pixel

(Adding an extra variable to any of the above changes their colour)

[27] [n ; t ; s gn change Font

- n = Font number
  t = Type (Bold, Italic etc.)
  s = Size

April 1994

WEEK 16

18 Monday
DAY (108-257)

19 Tuesday
DAY (109-256)

20 Wednesday
DAY (110-255)

21 Thursday
DAY (111-254)

22 Friday
DAY (112-253)

23 Saturday
DAY (113-252)
St. George's Day

24 Sunday
DAY (114-251)
3rd Sunday after Easter

Memorandum

April/May 1994

WEEK 17

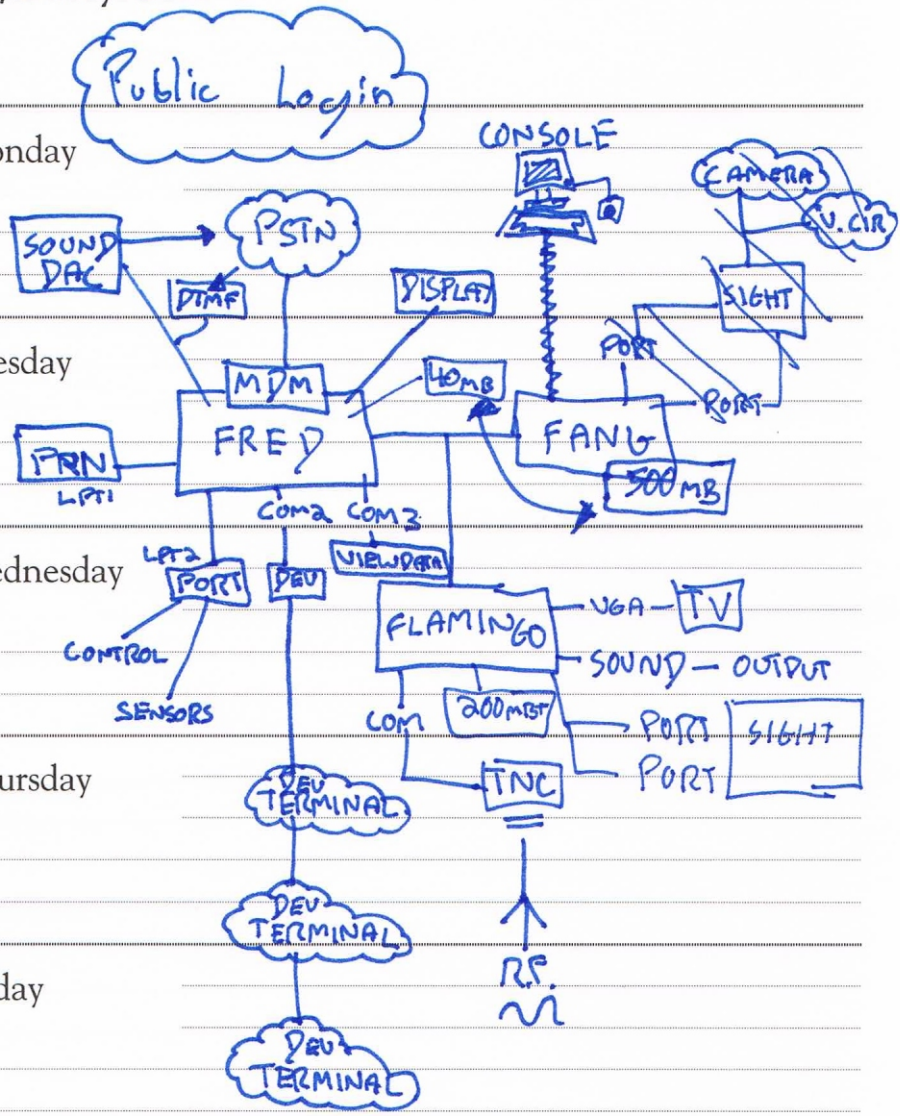
25 Monday
DAY (115-250)

26 Tuesday
DAY (116-249)

27 Wednesday
DAY (117-248)

28 Thursday
DAY (118-247)

29 Friday
DAY (119-246)



30 Saturday
DAY (120-245)

1 Sunday
DAY (121-244)
4th Sunday after Easter

Memorandum

0 2 @ 500MB HD - 300
0 1 @ 486 33 + BOARD - 260
0 4MB Memory - 120
0 1 @ V42 BIS Modem Card - 90

$700 to complete

PSU - 50W @
needed = 40a w.

May 1994

WEEK 18

2 Monday

DAY (122-243)
Bank Holiday UK

3 Tuesday

DAY (123-242)

4 Wednesday

DAY (124-241)

5 Thursday

DAY (125-240)

6 Friday

DAY (126-239)

7 Saturday

DAY (127-238)

8 Sunday

DAY (128-237)
Rogation Sunday
Mother's Day USA CDN

Memorandum

May 1994

WEEK 19

MCD INTERACTIVE V1.0

9 Monday
DAY (129-236)

10 Tuesday
DAY (130-235)

Define Object.

[1] All objects are plotted around an origin of (0,0) and may consist of any number of the following elements

11 Wednesday
DAY (131-234)

1 circle
2 Line
3 Box
4 Filled Box

12 Thursday
DAY (132-233)
Ascension Day

5 Dot/Pixel
6 Random Pixels (Rn), area n1 9n2
7 Area Fill
8 Shade effect 'A'
9 Shade effect 'B'

13 Friday
DAY (133-232)

10 Text (normal System font)
11 Square / elipse shaded
12 Elipse (circle + Elip Number)

14 Saturday
DAY (134-231)

If algorithm is used ends in CHR 1 else CHR 0 terminates.

15 Sunday
DAY (135-230)
Sunday after Ascension

Movement [2]

Objects can be assigned an algorythm which is used against a system clock to move the object.

Individaul object elements may also be assigned algorythms to enable some animation to take place.

[3] Set origin (sets next objects position)

[2 Byte] [2 Byte]

[4] Set object Number [1 Byte]
Must be set before an object is drawn.

[5] Set Mode Resolution [1 Byte]

7 = 320, 200 x16
8 = 640, 200 x16
9 = 640, 340 x16
12 = 640, 440 x16
13 = 320, 200 x255

0 = Text mode 1 (80x25)
1 = Text Mode 2 (40x25)
2 = Text Mode 3 (80x50)

May 1994

WEEK 20

16 Monday

DAY (136-229)
Shavuoth

17 Tuesday

DAY (137-228)

18 Wednesday

DAY (138-227)

19 Thursday

DAY (139-226)

20 Friday

DAY (140-225)

21 Saturday

DAY (141-224)
Armed Forces Day [USA]

22 Sunday

DAY (142-223)
Pentecost (Whit Sunday)

Memorandum

Algo: Per Obstacle Element.

Timing Information 10sec / n

Increasement /10(-127 → 127) per tick

Reference x, y co-ord

Algo 1 = x + Inc
Algo 2 = x - Inc
Algo 3 = x / Inc
Algo 4 = x * Inc
Algo 5 = x + n * Cos (Inc)
Algo 6 = x + n * Sin (Inc)
Algo 7 = x + n * Cos (Inc/10)
Algo 8 = x + n * Sin (Inc/10)

May 1994

WEEK 21

23 Monday

DAY (143-222)
Victoria Day,
Holiday CDN

24 Tuesday

DAY (144-221)

25 Wednesday

DAY (145-220)

26 Thursday

DAY (146-219)

27 Friday

DAY (147-218)

28 Saturday

DAY (148-217)

29 Sunday

DAY (149-216)
Trinity Sunday

Memorandum

May/June 1994

WEEK 22

30 Monday

DAY (150-215)
Bank Holiday [UK]
Memorial Day, Holiday [USA]

31 Tuesday

DAY (151-214)

1 Wednesday

DAY (152-213)

2 Thursday

DAY (153-212)

3 Friday

DAY (154-211)

4 Saturday

DAY (155-210)

5 Sunday

DAY (156-209)
2nd Sunday after Pentecost

Memorandum

June 1994

WY-60 Escape Codes

WEEK 23

6 Monday
DAY (157-208)
Holiday (IRL)

CHARACTER SETS

ESC c D - Select Primary set
ESC c E - Select Secondary set
ESC c B Bank - Define Primary set

7 Tuesday
DAY (158-207)

ESC c C Bank - Define Secondary set
ESC c @ Bank, Set - Load soft Bank with char set
ESC c ? Bank - Clear soft Bank

8 Wednesday
DAY (159-206)

ESC c A Bank, pp b...b ^Y
(Define and load soft soft)
ESC e N - Auto load soft, off
ESC e O - Auto load soft, on
ESC c m codepage - Select codepage

9 Thursday
DAY (160-205)

Graphics Characters

ESC H Idraw - Display graph chr
ESC H CTRL B - Line draw graphics mode on

10 Friday
DAY (161-204)
Islamic New Year (1415)

ESC H CTRL C - Line draw mode off

11 Saturday
DAY (162-203)

Selecting Personalities
ESC " space - enhance mode on
ESC " ! - enhance mode off

12 Sunday
DAY (163-202)
3rd Sunday after Pentecost

ESC " personality - select it

Memorandum

Little 'w'

/ Display Memory / Split screen

Esc w length - Divide memory into pages
Esc w B - Disp Prev. Page
Esc w C - Disp Next Page
Esc w page - Disp specified page

Esc e A line - split screen Horz (simple)
Esc e 1 line - splitscreen Horz Clr (simple)
Esc e C line - split Horz (Adjust)
Esc e 3 line - split Horz Clr (Adjust)

Esc J - Activate upper win
Esc 3 - Activate lower win
Esc J - Activate other win
Esc K - Activate other win

Esc e P - Lower Horz split
Esc e R - ~~Raise~~ Raise Horz split

Esc w E - Roll up window in page
Esc w F - Roll down window in page

Esc e @ - Redizine screen as 1 window
Esc e Ø - Redizine screen as 1 window clr.

Esc d(- Autodrag cursor off
Esc d) - Autodrag cursor on

Esc w D - Reposition workspace.

June 1994

WEEK 24

13 Monday
DAY (164-201)

23 = Blank
22 = Full square ?

14 Tuesday
DAY (165-200)
Flag Day [USA]

15 Wednesday
DAY (166-199)

16 Thursday
DAY (167-198)

17 Friday
DAY (168-197)

18 Saturday
DAY (169-196)

19 Sunday
DAY (170-195)
4th Sunday after Pentecost
Father's Day [UK] [USA] [CDN]

Memorandum

June 1994

WEEK 25

20 Monday
DAY (171-194)

21 Tuesday
DAY (172-193)
Longest Day

22 Wednesday
DAY (173-192)

23 Thursday
DAY (174-191)

24 Friday
DAY (175-190)
St. Jean-Baptiste Day CDN
(Québec)

25 Saturday
DAY (176-189)

26 Sunday
DAY (177-188)
5th Sunday after Pentecost

Memorandum

June/July 1994

WEEK 26

27 Monday
DAY (178-187)

28 Tuesday
DAY (179-186)

29 Wednesday
DAY (180-185)

30 Thursday
DAY (181-184)

1 Friday
DAY (182-183)
Canada Day, Holiday [CDN]

2 Saturday
DAY (183-182)

3 Sunday
DAY (184-181)
6th Sunday after Pentecost

Memorandum

July 1994

WEEK 27

4 Monday
DAY (185-180)
Independence Day, Holiday [USA]

5 Tuesday
DAY (186-179)

6 Wednesday
DAY (187-178)

7 Thursday
DAY (188-177)

8 Friday
DAY (189-176)

9 Saturday
DAY (190-175)

10 Sunday
DAY (191-174)
7th Sunday after Pentecost

Memorandum

July 1994

WEEK 28

11 Monday
DAY (192-173)

12 Tuesday
DAY (193-172)
Holiday NI

13 Wednesday
DAY (194-171)

14 Thursday
DAY (195-170)

15 Friday
DAY (196-169)

16 Saturday
DAY (197-168)

17 Sunday
DAY (198-167)
8th Sunday after Pentecost

July 1994

WIX-60 Escape Codes

WEEK 29

18 Monday
DAY (199-166)

ESC c T = Default all modes
ESC c V = Save setup modes 2 non vol
ESC c W = Save tests & setup 2 non vol
ESC c R = Power On reset

19 Tuesday
DAY (200-165)

01734
- Wyse Connect

20 Wednesday
DAY (201-164)

80437 personalities
80338 codes

21 Thursday
DAY (202-163)

22 Friday
DAY (203-162)

23 Saturday
DAY (204-161)

24 Sunday
DAY (205-160)
9th Sunday after Pentecost

Memorandum

July 1994

WEEK 30

25 Monday
DAY (206-159)

26 Tuesday
DAY (207-158)

27 Wednesday
DAY (208-157)

28 Thursday
DAY (209-156)

29 Friday
DAY (210-155)

30 Saturday
DAY (211-154)

31 Sunday
DAY (212-153)
10th Sunday after Pentecost

'If' Conditional Statements

```
If { value1 > = Value2 }
@)
    'Condiction True'

@)
Else 'optional condition false'
@)
    'condiction False'
@)
```

August 1994

WEEK 31 TRE940-W

1 Monday
DAY (213-152)
Holiday SCOT IRL

Network Protocol for WAN communication using the extended TRE940 standard.

2 Tuesday
DAY (214-151)

Header - Preceeds all commands

H+ = {27} {1 BYTE} {11 BYTES} {11 BYTES} {4 BYTES}

COUNTRY CODE | SERVICE ID | TERM ID | NETWORK No.

3 Wednesday
DAY (215-150)

eg. (44) - UK
Voyg. Tre. - Service ID
94-0000-001 - Term ID
FFF0 - Network ID (EPN)

H+ = 27 + H_1 + H_2

Tx PC | Rx PC

4 Thursday
DAY (216-149)

TRANSFER DATA BLOCK

REQUEST 01 | H+ {01} + {4 nB} + {L n2} + {4 Byte} + {4 B}

Length | checksum

5 Friday
DAY (217-148)

Maximum of 64K can be transfered at any one time between two computers. Above command requests allocation of space from the client. Buffer is the 'handle' to which this data will be referenced to.

6 Saturday
DAY (218-147)

ACK 02 | H+ {02} + {ERR}

0 = OK TO TRANSFER (Buffer allocated & next transfer from PC)

7 Sunday
DAY (219-146)
11th Sunday after Pentecost

>0 = Error. Std Error.

TRE940-W

Transfer Data Block - Send Block
03

H+ {03} + $$$$$

Sends entire Data Block, and waits for ACK CS from Client. Is To. of more than 20 secs - About Transfer.

ACK CS Acknowledge Check Sum
04

H+ {04} + {FLAG}
L 00 = OK (Perfect send)
>0 = 1 = Re send
2 = Do Not Re send.

Buffer Manipulation
05

H+ {05} + {1 Byte} + {Ln1} + {Ln2} + {St1} + {St2} + {4 Byte}
Func. — Buffer to

Ln = Length to use, St = start Position

Func = 1 - Fixed Window (create Window)
Func = 2 - Update Window
Func = 3 - Store File (create file)
Func = 4 - Update File

{4 Byte} Req Buff
0 is No Request

Func = 5 - Play Voice Data
Func = 6 - Play MKV Data in Window
Func = 7 - Email
Func = 8 -
Func = 9 -

Func = 10 - System TreeServe Direct Access

August 1994

WEEK 33

TRE-940 - W

Bugger TX to {4 Byte}

15 Monday
DAY (227-138)

H+ {06} + {4 Byte} + Rqst Bugger
Bugger Rq

(6)

Request Bugger to TX from Host.
In all transactions the first 255 Buggers are Reserved. Details below.

16 Tuesday
DAY (228-137)

Buggers Used By System

17 Wednesday
DAY (229-136)

01 - 127 - OPEN FILE LIST / PATHS
(FILES/PATHS BLOCK)

18 Thursday
DAY (230-135)

125 - 167 - Window Display
information.
(WINDOWS ACCESS BLOCK)

19 Friday
DAY (231-134)

H+ {07} + {1 Byte} Get Free Bugger
Type

Type 1 = File List
Type 2 = Window

20 Saturday
DAY (232-133)

Gets next free System bugger avalible to host.

21 Sunday
DAY (233-132)
13th Sunday after Pentecost

Memorandum

TRE940.W

Files / Paths Block 00 - OFFSET

8 Byte
ID = 255 + 0 + FILPAT

H+ - Sources of Files / Paths

11 Byte - Remote's Disk Label

1 Byte - Remote's Disk Drive (0 = Default)

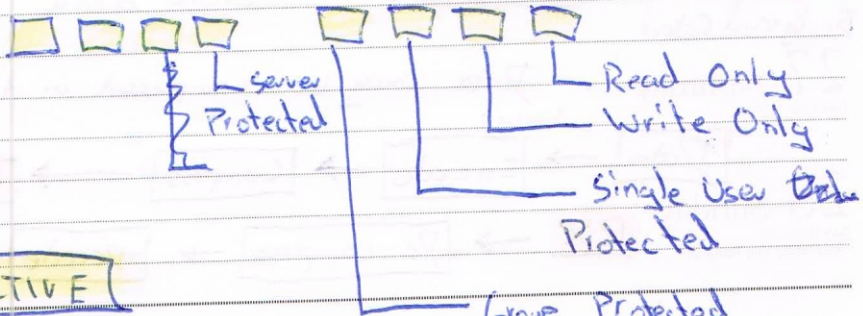
64 Byte - Current Path

2 Byte - Total Items

$$$$$

- 1 Byte - Attribute (DOS)
- 8 Byte + 3 Byte - Name
- 1 Byte - Attribute (TREESERVE)

Attributes used by TreeServe



1 = ACTIVE

August 1994

TREG40 - W

WEEK 34

22 Monday
DAY (234-131)

Picture (UN-COMPRESSED) data format

For the Displaying of Bitmapped images in new windows, or as the 'seed' to an MCU datastream.

23 Tuesday
DAY (235-130)

This format is for 2D, 255 Colour images. More Colours can be added by multiplying 255 by the FLAGS REG

Data Block

24 Wednesday
DAY (236-129)

- {10 Byte} - FLAGS (colour multiplier)
- {4 Byte} - X Resolution
- {4 Byte} - Y Resolution

25 Thursday
DAY (237-128)

{1 Byte} - Total Colours (255, 16, ~~4~~, 2)

{$$$$$$} - RGB Palette - One byte

26 Friday
DAY (238-127)

{~~10~~ 40 Bytes} - Title of Picture

{1 Byte} + {$$$$$$} - Picture data
Background Colour

27 Saturday
DAY (239-126)

Data may be compressed in the following way:

Data → Encode → Compress → TX

28 Sunday
DAY (240-125)
14th Sunday after Pentecost

RX → De-compress → Decode → Data

Memorandum

TREGHO-W

[255] + [255] = Encode Mode. Otherwise data is sent 'Raw' to screen.
+ [MODE] (1 Byte)

NB. MODE 0 = Raw Mode (2×255 = ~~[illegible]~~) (255+255+255)

Mode 1 (Full colour encode)

{1 Byte} = Length + {1 Byte} = Data (colour)

Mode 2 (16 colour encode) - (Full)

({16 Bytes} 0 - 15 colour Map)

{1 Byte} + {Half Byte} {Half Byte} + {1 Byte Ln}

Length

Mode 3 (16 Colour encode) - (Half)

3] {Half Byte} + {Half Bytes}
Length Colour

(Uses mapping as above if 255 colour Pic)

Mode 4 (2 colour encode) - (Full)

{1 Byte} - 0 map {1 Byte} - 1 map

{127 max} + {1 BIT}
Length Colour

Aug/Sept 1994

TRE940-W

WEEK 35

29 Monday
DAY (241-124)
Holiday UK (except Scotland)

Mode 5 (2 colour encode) - Half

{1 Byte} 0 Map {1 Byte} 1 Map

30 Tuesday
DAY (242-123)

{3 BITS + 1BIT} + {3 Bits + 1bit}

Length Colour Length Colour

31 Wednesday
DAY (243-122)

Mode 6 Detail (Pattern Encode) FULL (COLOUR)

{1 Byte} {1 Byte x Ln} {1 Byte}

Length of Pattern Colour Set up Mapping Map to store as

1 Thursday
DAY (244-121)

Mode 7 Detail (Pattern Encode) (16 colour)

{0-15} Pallette Map

2 Friday
DAY (245-120)

{1 Byte} {1 Byte x Ln} {1 Byte}

Length of Pattern

{4 Bit} {4 Bit}

Half Byte

Map to store as

3 Saturday
DAY (246-119)

4 Sunday
DAY (247-118)
15th Sunday after Pentecost

TRE940-W

Mode 8 Detail (Pattern Encode) - 2 colour

{Byte 1} 0 - Pal. Map {Byte 1} 1 - Pal map

{1 Byte} + {8x1 Bit} + {1 Byte}
Length X Length store for Pattern

Mode 9 Display Pattern Store

{Byte 1} = Map to display

Displays a stored Map at current position. Cursor is then increased by Patterns length.

Mode 10 Line Sync

Fills rest of line with current colour and places cursor at X=0, Y=Y+1. Line sync automatically occurs when X = Max value.

Mode 11 Home

Places cursor at X=0, Y=0

Mode 12 Move Cursor

{4 Byte} X POS {4 Byte} Y POS

WEEK 36

TRF940-W

5 Monday
DAY (248-117)
Labor Day, Holiday USA CDN

Mode 13 — 0=Cancel, 1=1bit

{1 Byte} total expansion = (~~[illegible]~~ pixels)

Expand cursor downwards.

Every new pixel ploted to screen will be ploted on (n) pixels below the curent pixels cursor.

6 Tuesday
DAY (249-116)
Rosh Hashanah

Mode 14 Copy line / block

{4 Byte} X pos | {4 Byte} Y pos | {4 Byte} X Len | {4 Byte} Y len

7 Wednesday
DAY (250-115)

{4 Byte} Copy to X | {4 Byte} Copy to Y

Copys an area of already received material to another part of the display.

8 Thursday
DAY (251-114)

Mode 15 Rotate Line

X Pos {4 Byte} | {4 Byte} X Rot + | [illegible]

9 Friday
DAY (252-113)

Rotates line by (n) pixels added to the 'X' plane. Data leaving Right of screen is displayed on the left of the line.

10 Saturday
DAY (253-112)

Mode 16 Skip Pixels

{4 Byte} No. to skip. — X ▶ Moves in 'X' plane only.

11 Sunday
DAY (254-111)
16th Sunday after Pentecost

Used primarily in MCU, ~~[illegible]~~ when sending difference only information after the 'seed' frame has been received.

TRE 940 -w

MODE 17 3D Support Services 1

{1 Byte} $$n...
Sub-Command

1 = Copy entire frame to Right Hand frame. (2D)
(sent at end of transmission for 2D images)
(Used in 3D imaging, MCW Mode 1)

2 = Copy block in frame to Right Hand frame.

{4 Byte}	{4 Byte}	{4 Byte}	{4 Byte}
X POS (LH)	Y POS (LH)	X LEN	Y LEN

{4 Byte}	{4 Byte}
X POS (RH)	Y POS (RH)

3 = Copy Item within Colour range to Right H frame.

{4 Bytes}	{4 Byte}	{4 Byte}	{4 Byte}
X POS LH	Y POS LH	X LEN	Y LEN

{4 Byte}	{4 Byte}	{4 Byte}	{4 Byte}
X POS RH	Y POS RH	COLOUR MIN	COLOUR MAX

MODE 18 Aspect Ratio Select Services

{1 Byte}	{4 Byte}
FLAGS	X POS

0 = 4:3
1 = 16:9

→ Movement in 4:3 display for 16:9 image

September 1994

WEEK 37

TRE940-W

12 Monday
DAY (255-110)

Treeserve internal functions

Ø = ACK + {1 Byte} + {1 Byte}

Ref

Err = Ø => No Error

13 Tuesday
DAY (256-109)

1 OPEN FILE

{1 Byte} Reference

{4 Byte} Buffer with files details on.

14 Wednesday
DAY (257-108)

{4 Byte} Offset. to FILES/PATHS block

{4 Byte} Buffer for File I/O

15 Thursday
DAY (258-107)
Yom Kippur

2 Read File

{1 Byte} Reference

{8 Byte} Position (Ø = From last)

16 Friday
DAY (259-106)

{4 Byte} Length (64K max)

17 Saturday
DAY (260-105)

3 Write File

{1 Byte} Reference

18 Sunday
DAY (261-104)
17th Sunday after Pentecost

{8 Byte} Posistion (Ø = From last

{4 Byte} Length

4 CLOSE FILE (frees up Buffer)

{1 Byte} Reference

5 Set Drive / Path for DIR to Host

{1 Byte} Ref

{4 Byte} Buffer with File/Path Block

{4 Byte} Offset to File / Path Block

6 Erase File(s)

{1 Byte} Ref

{4 Byte} Buffer with File/Path Block

{4 Byte} Offset to File/Path Block

7 Copy file (Does not erase original)

{1 Byte} Ref

Source file

{4 Byte} Buff {4 Byte} Off

{4 Byte} Buff {4 Byte} Off

Target Path / Drive

WEEK 38

File Attributes

19 Monday
DAY (262-103)

List of all files which have special TreeServe Attributes associated with them, is stored in the same path as it's files.

20 Tuesday
DAY (263-102)
Succoth

FILENAME - CHR$(255) + "TSAT" + CHR$(1) + CHR$(255) + "TS" + CHR$(255)

NB: (127) CHR can be used to screen files from some lists.

21 Wednesday
DAY (264-101)

Block structure

{2 BYTE} {255} + {02} (ID)

$b... {11 Byte} (Filename)

22 Thursday
DAY (265-100)

{1 Byte} Attribute

{11 Byte} Owner (or creator)

23 Friday
DAY (266-99)

24 Saturday
DAY (267-98)

25 Sunday
DAY (268-97)
18th Sunday after Pentecost
EC Summer Time ends

For group owned files the 11 Byte 'Owner' ID is the name of a created 'group ID' on that files server / terminal.

A list of Terminals which are part of this group are held in the Root + \ System path and can only be changed by the Terminal / server which holds the file. (No remote access).

27)

Group List File

FILENAME - CHR$ (255) + 'TSGP' + CHR$ (127) + CHR$ (255) + '.TS' + CHR$ (255)

Block Structure

{2 Byte} {255} + {03} (ID)

~~[illegible]~~

~~[illegible]~~

~~[illegible]~~

{11 Byte} Group ID

{2 Byte} No of members

{11 Byte} Term ID of Member (n)

Sept/Oct 1994

WEEK 39

26 Monday
DAY (269-96)

27 Tuesday
DAY (270-95)

28 Wednesday
DAY (271-94)

29 Thursday
DAY (272-93)

30 Friday
DAY (273-92)

1 Saturday
DAY (274-91)

2 Sunday
DAY (275-90)
19th Sunday after Pentecost

Memorandum

October 1994

WEEK 40

3 Monday
DAY (276-89)

Used to find frame if 'sync' is lost or file is corrupt.

4 Tuesday
DAY (277-88)

4 Byte

Run in code
(00, 255, 254, 00)

5 Wednesday
DAY (278-87)

2 Byte — X pos

2 Byte — Y pos

6 Thursday
DAY (279-86)

1 Byte — Cmnd

1 Byte — Function

2 Byte — Length (Length of Data which follows)

7 Friday
DAY (280-85)

Commands

K		F	
01	–	00	(PSET IMAGE) + Leng
01	–	01	(XOR IMAGE)
01	–	02	(OR IMAGE)
01	–	03	(AND IMAGE)

8 Saturday
DAY (281-84)

02 – 00 (MOVE AREA)

9 Sunday
DAY (282-83)
20th Sunday after Pentecost

~~Dat~~ (2 Byte — X2 | 2 Byte — Y2 | 2 Byte — MX | 2 Byte — MY

03 - 00 (Text output)

DAT: 1 Byte - Colour
1 Byte - Background Colour (255 = Transparent)
1 Byte - X size
1 Byte - Y size
1 Byte - Use Font* (0 = No font)
1 Byte - Effect 1
1 Byte - Effect 2
1 Byte - Sub Text function (not Used)

{1 Byte} - String Length

{$$$} - ASCII Text String for Output.

October 1994

WEEK 41

10 Monday
DAY (283-82)
Columbus Day Observed, Holiday [USA]
Thanksgiving Day, Holiday [CDN]

04-00 Play Sound

DAT = Digitised Sound Data.

11 Tuesday
DAY (284-81)

04-01 Repeat Current Sound

04-02 Sound Off

04-03 Store Sound to Buffer
DAT {1 Byte} = Buffer

12 Wednesday
DAY (285-80)

04-04 Retreive Buffer
DAT {1 Byte} = Buffer.

13 Thursday
DAY (286-79)

14 Friday
DAY (287-78)

15 Saturday
DAY (288-77)

16 Sunday
DAY (289-76)
Last Sunday after Pentecost

05-00 ▶ Interactive Services

01 - Key Press

DAT = KEY TO SCAN ~~(255 = RESET)~~

+ {2 Byte} (0 = RESET)

<> DIRECTION OF SYNC TO CONTINUE WITH.

02 - SCREEN SELECT

DAT = {2 Byte} {2 Byte} {2 Byte} {2 Byte}

X_1 X_1 X_2 Y_2

~~FFFF = RESET~~

{2 Byte} (0 = RESET)

<> DIRECTION OF SYNC TO CONTINUE WITH

03 - FAST OPTION / WINDOW BUTTON

DAT = {1 Byte} option Number

~~{1 Byte}~~ {1 Byte} $ Length

{2 Byte} (0 = RESET)

<> DIRECTION OF SYNC TO CONTINUE WITH

{$$$$} STRING FOR OPTION NAME

Loop Statements

17 Monday
DAY (290-75)

~~if~~ ~~stm~~ ~~ifn~~ Value1 < Value2 .and. [1]=n
(.inc. Value [n])
(.dec. Value [n])

18 Tuesday
DAY (291-74)

The Tsh has no proper 'for' loop statements. Instead the 'If' and 'Ifn' commands are used as loops by placing an increasement statement, such as:

19 Wednesday
DAY (292-73)

```
If  a < 10  .Inc. a [1]
{
  Prn a
  Prn 'and still counting..'
}
```

20 Thursday
DAY (293-72)

The above would execute the instructions while the condition (a<10) remains true. Multiple increase/decrements may be used such as:

21 Friday
DAY (294-71)

```
Ifn  a>0 .and. b<10 .Inc. b [1]
.dec. a [1] .d.a [0] .d. b [1
```

The additional commands for the if statements include:

22 Saturday
DAY (295-70)

.Inc. - Increase value
.dec. - decrease value
.d.* - ~~set value~~ initialise value

23 Sunday
DAY (296-69)
9th Sunday before Christmas
British Summer Time ends

~~Must appear at the~~ Must appear at the end of the statement.

Conditional Statements

If Value1 < Value2 .and. [2] = n

- Searches for the next '{' for the 'true' condition and if false will either goto the next '{{' in the current '{-3' segment or continue execution after the end of the current segments '3'.

Ifn Value1 < Value2 .and. [2] = n

- Searches for the next '{' for the 'true' condition, and if false will continue execution at the command immediatly after the 'Ifn'. With a true condition execution continues at the end of the '{-3' segment.

'{' = Begining of 'True' Segment
'{{' = Begining of nested 'False' Segment
'3' = End of Segment

'{' = $func 2, func 1
'{{' = $func 2, func 2
'3' = $func 2, func 3

October 1994

WEEK 43

24 Monday
DAY (297-68)
United Nations Day

25 Tuesday
DAY (298-67)

26 Wednesday
DAY (299-66)

27 Thursday
DAY (300-65)

28 Friday
DAY (301-64)

29 Saturday
DAY (302-63)

30 Sunday
DAY (303-62)
8th Sunday before Christmas
Daylight Saving Time ends [USA] [CDN]

Loop Functions

Tsh has no 'for' statement, instead 'if' statments are used with certain special extensions which make loop control possible.

```
If .d. n [0] .inc. n [1] n < [10]
   define      counter   condition
```

Note that the order must be ~~in the~~ the same as shown above, but multiples of either any of the 3 main areas of the function are possible.

Also note that you may 'nest' loop functions as shown below:

```
If .d. n [0] .inc. n [1] n < [10]
{
If .d. p [5] .dec. p [1] p > [0]
{ ~~prn n, p~~
  prn n, p
}
}
```

You may nest upto 24 individual loops at any one time. Each '}' will return control to the function in use previously.

Ifn statement may also be used for loops.

WEEK 45

Tnet Switching Network.

7 Monday
DAY (311-54)

8 Tuesday
DAY (312-53)
Election Day [USA]

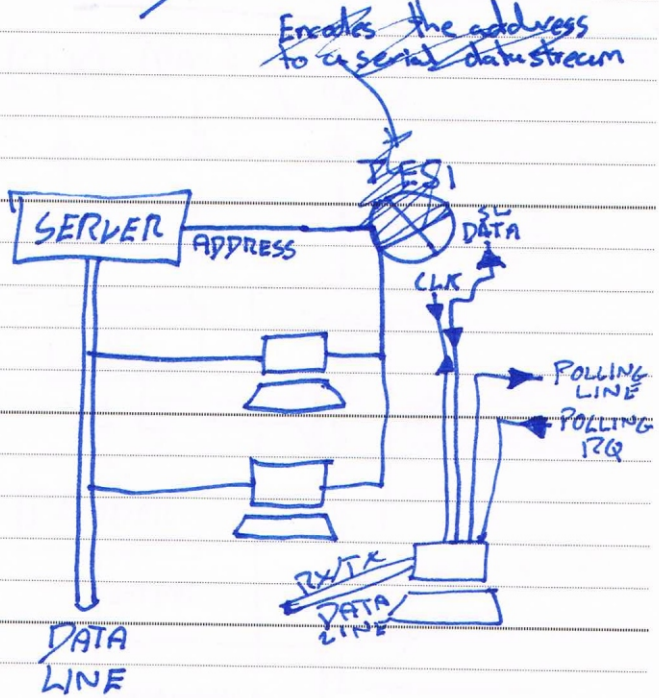
9 Wednesday
DAY (313-52)

10 Thursday
DAY (314-51)

11 Friday
DAY (315-50)
Veterans' Day, Holiday [USA]
Remembrance Day, Holiday [CDN]

12 Saturday
DAY (316-49)

13 Sunday
DAY (317-48)
6th Sunday before Christmas
Remembrance Sunday [UK]

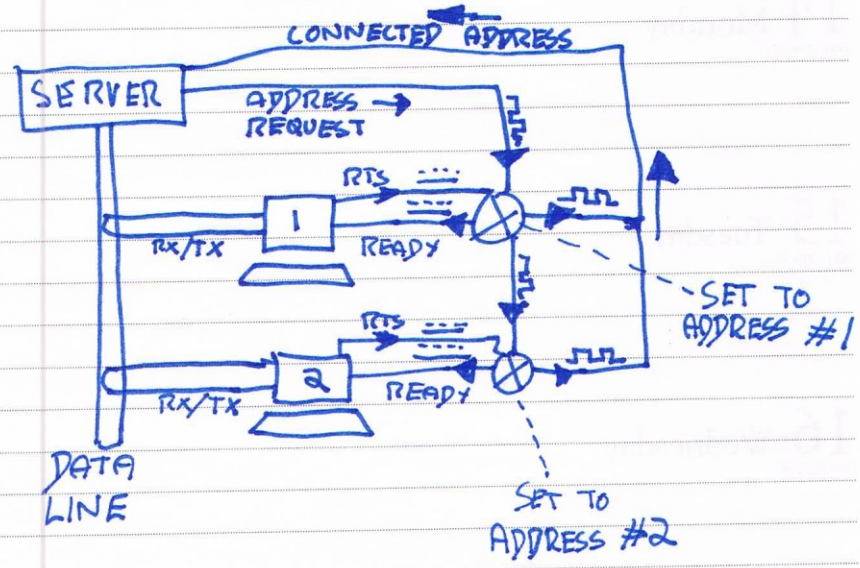


FOR A TERMINAL REQUESTING TO SEND

① Check to see if 'polling Line' is low
② Pull up Polling Line and wait to be sent an address with the 'polling request' line low.
(WAIT 10ms) ③ Send your address back, keeping polling request line high.
(WAIT 10ms) ④ On receiving your address again with the 'polling request' line low send all data from terminal, not dropping 'polling line' until RTS goes low.

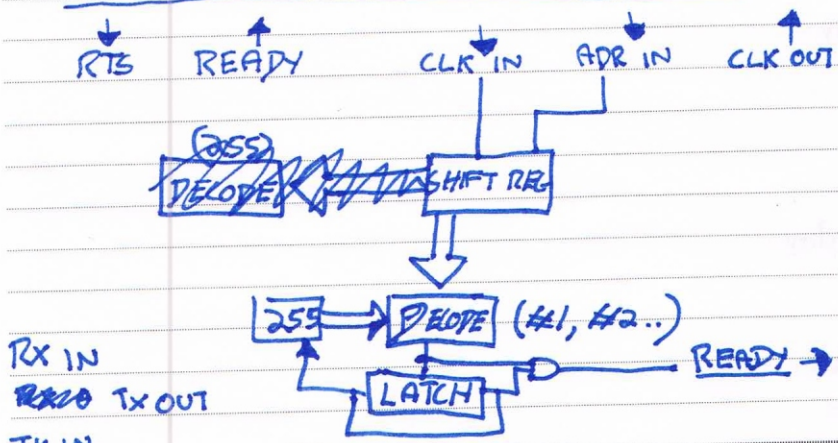
● Note: Drop 'polling Line' after 2 secs and the CTS is not valid

TAS (TWIN ADDRESS SWITCHING)



- PROVIDES UP TO 254 (+1 IDLE CONDITION) CONECTIONS TO TERMINALS
- ENSURES CONNECTION TO THE RIGHT TERMINAL

BLOCK DIAGRAM OF A TAS SWITCH.



November 1994

WEEK 46

14 Monday
DAY (318-47)

15 Tuesday
DAY (319-46)

16 Wednesday
DAY (320-45)

17 Thursday
DAY (321-44)

18 Friday
DAY (322-43)

19 Saturday
DAY (323-42)

20 Sunday
DAY (324-41)
5th Sunday before Christmas

Memorandum

November 1994

WEEK 47

21 Monday
DAY (325-40)

&Terminal Server - On During Working hours only

&Gore - On when in use only

22 Tuesday
DAY (326-39)

&Access - Always on, each unit takes a 1 hour break in early hours.

23 Wednesday
DAY (327-38)

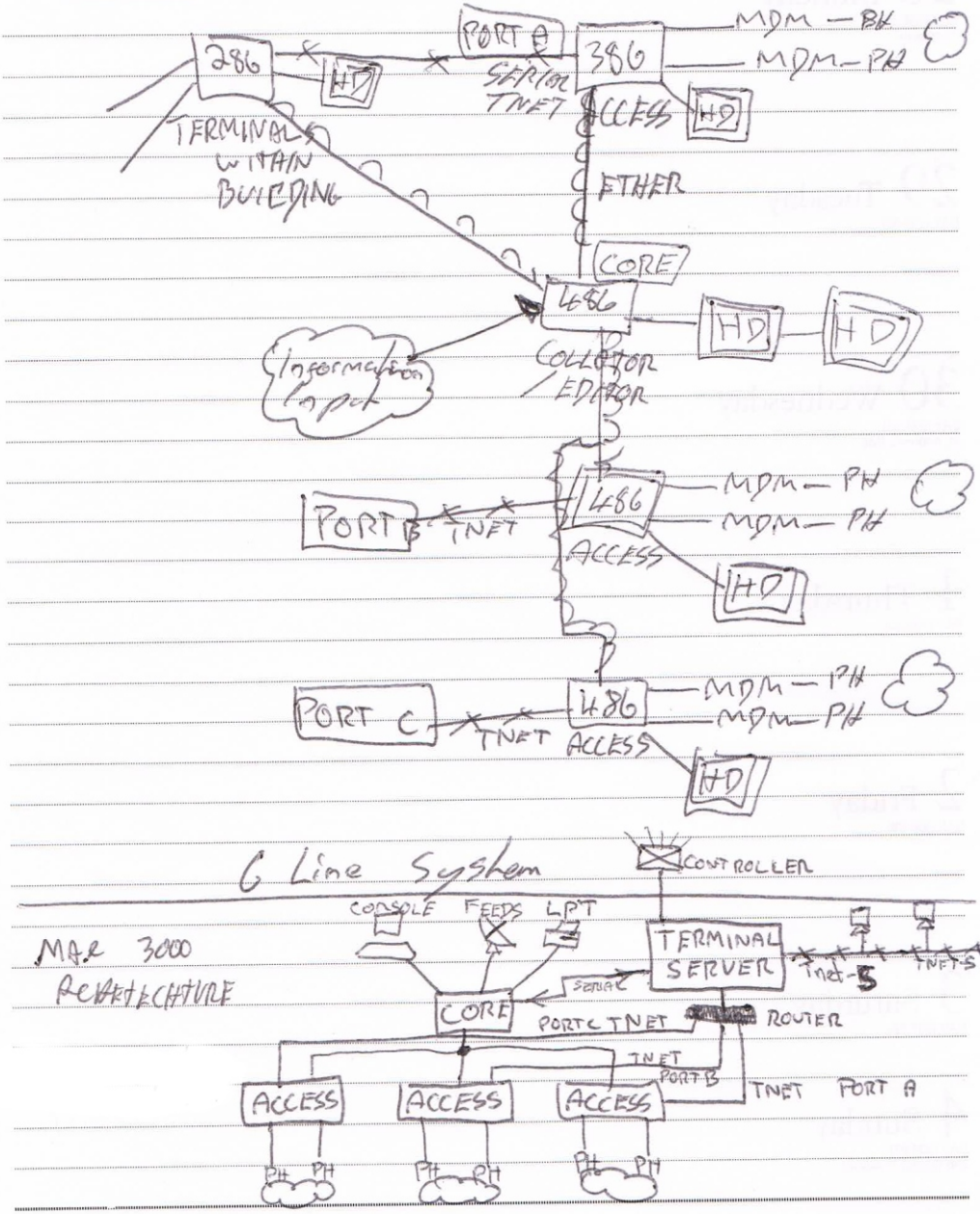
24 Thursday
DAY (328-37)
Thanksgiving Day [USA]

25 Friday
DAY (329-36)

26 Saturday
DAY (330-35)

27 Sunday
DAY (331-34)
1st Sunday in Advent

Memorandum


286
HD
PORT A
SERIAL
TNET
386
MDM — PH
MDM — PH
ACCESS
HD
TERMINAL WITHIN BUILDING
ETHER
CORE
486
HD
HD
Information input
COLLECTOR EDITOR
486
MDM — PH
MDM — PH
PORT B
TNET
ACCESS
HD
486
MDM — PH
MDM — PH
PORT C
TNET
ACCESS
HD
C Line System
CONTROLLER
CONSOLE
FEEDS
LPT
TERMINAL SERVER
Tnet-B
TNET-S
MAR 3000
ARCHITECTURE
SERIAL
CORE
PORT C TNET
ROUTER
TNET
PORT B
TNET
PORT A
ACCESS
ACCESS
ACCESS
PH
PH
PH
PH
PH
PH

Nov/Dec 1994

WEEK 48

28 Monday
DAY (332-33)
Chanucah

29 Tuesday
DAY (333-32)

30 Wednesday
DAY (334-31)
St. Andrew's Day

1 Thursday
DAY (335-30)

2 Friday
DAY (336-29)

3 Saturday
DAY (337-28)

4 Sunday
DAY (338-27)
2nd Sunday in Advent

Memorandum

December 1994

WEEK 49

5 Monday
DAY (339-26)

6 Tuesday
DAY (340-25)

7 Wednesday
DAY (341-24)

8 Thursday
DAY (342-23)

9 Friday
DAY (343-22)

10 Saturday
DAY (344-21)

11 Sunday
DAY (345-20)
3rd Sunday in Advent

Memorandum

NE2000 TCP Packet Driver

Default address

ne2000 0x60 11 300

Hardware interrupt

December 1994

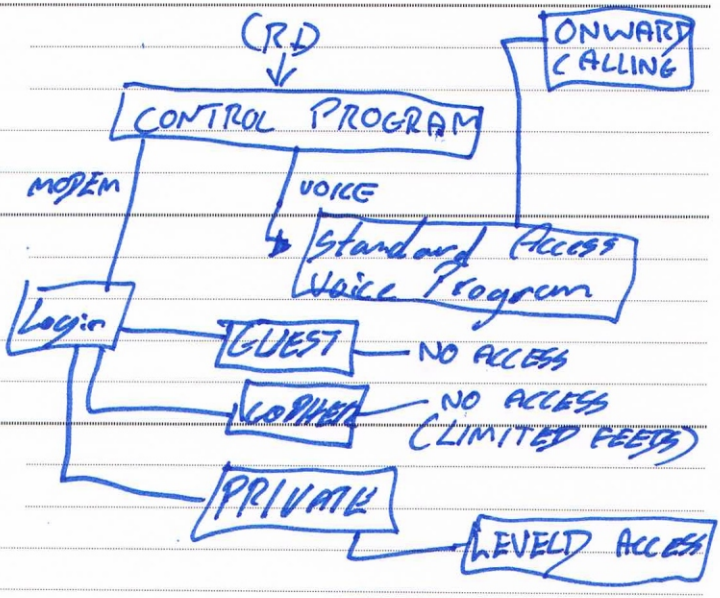
WEEK 50

PSTN Operation

12 Monday
DAY (346-19)

Modem → Control Program (RD)
Voice →

13 Tuesday
DAY (347-18)



14 Wednesday
DAY (348-17)

15 Thursday
DAY (349-16)

16 Friday
DAY (350-15)

17 Saturday
DAY (351-14)

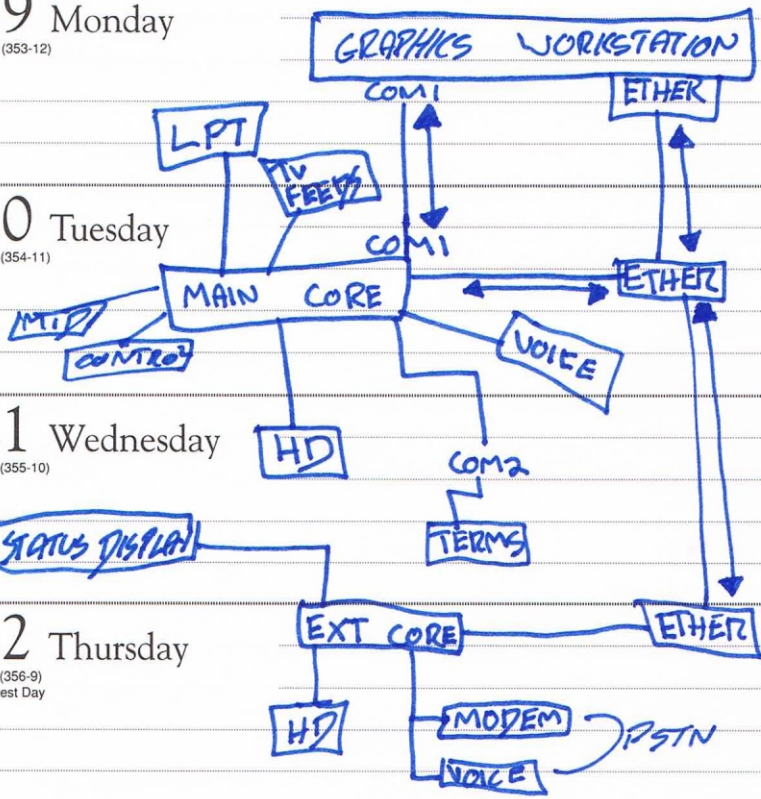
18 Sunday
DAY (352-13)
4th Sunday in Advent

Memorandum

December 1994

WEEK 51

SERVER ARCH.



19 Monday
DAY (353-12)

20 Tuesday
DAY (354-11)

21 Wednesday
DAY (355-10)

22 Thursday
DAY (356-9)
Shortest Day

23 Friday
DAY (357-8)

- Printer is redirected to COM port on (GW) to COM ports on the Main Core. Any data sent over this link which has no header, is sent directly to the Main Cores (Printer Spooler.)

24 Saturday
DAY (358-7)

- Applications on the (GW) can use the Main Cores HD as a network drive (with restrictions).

25 Sunday
DAY (359-6)
Christmas Day

Memorandum

INFORMATION, is transferred by either directly copying the file, or by sending a 'Request File'.

Eg. 1) Check whether @INUSE.DAT Exists.
2) Check Process ID of @INUSE.DAT is the same as @ACK.DAT. (CREATE A TIMOUT OF 1MIN FOR THIS) - To prevent a RESET/CRASH from causing system failure.
3) Send @INUSE.DAT which amongst other items will contain the following:

- Terminal ID of sender.
- Address and files path of where to send results
- Filename and path of the command file.
- Command Details

4) Send Command File.
5) Wait for @ACK.DAT file to be created.
6) Wait for results.

Dec/Jan 1994/95

WEEK 52

26 Monday

DAY (360-5)
Boxing Day, Holiday [UK] [IRL] [USA] [CDN]

27 Tuesday

DAY (361-4)
Holiday [UK] [IRL] [CDN]

28 Wednesday

DAY (362-3)

29 Thursday

DAY (363-2)

30 Friday

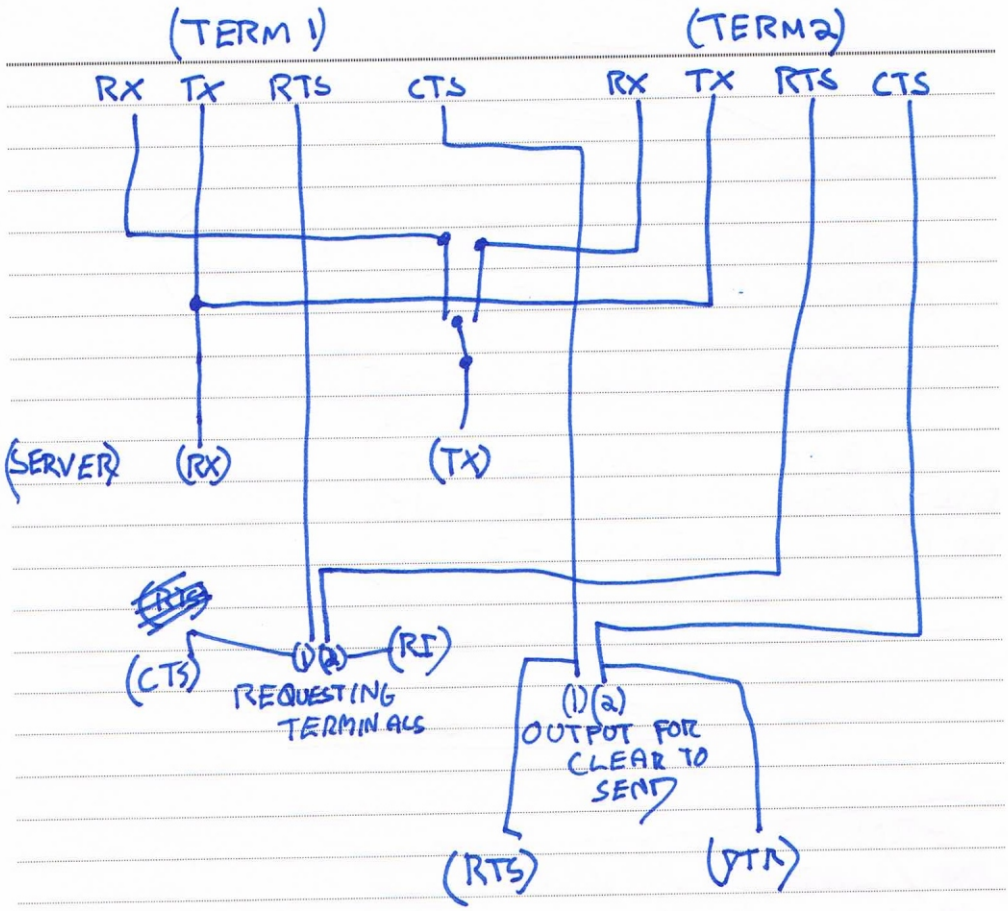
DAY (364-1)

31 Saturday

DAY (365)

1 Sunday

DAY (1-364)
1st Sunday after Christmas
New Year's Day

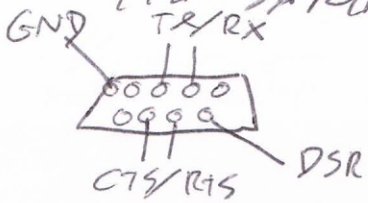


Alarm Keypad. — Set/unset signal

4.5v . 1.5v. — Movement PIR OUTPUT.

5v 1.5v.

9 PIN SERIAL PORT



2, 3 - TX, RX

7, 8 - CTS, RTS

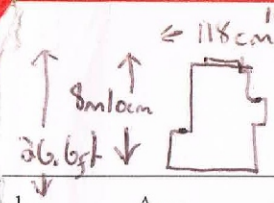
6 - DSR

5 - GND

4 - DTR

Year Planner

	Jan	Feb	March	April	May	June
Monday					1	
Tuesday					2	
Wednesday		1	1		3	
Thursday		2	2		4	1
Friday		3	3		5	2
Saturday		4	4	1	6	3
Sunday	1	5	5	2	7	4
Monday	2	6	6	3	8	5
Tuesday	3	7	7	4	9	6
Wednesday	4	8	8	5	10	7
Thursday	5	9	9	6	11	8
Friday	6	10	10	7	12	9
Saturday	7	11	11	8	13	10
Sunday	8	12	12	9	14	11
Monday	9	13	13	10	15	12
Tuesday	10	14	14	11	16	13
Wednesday	11	15	15	12	17	14
Thursday	12	16	16	13	18	15
Friday	13	17	17	14	19	16
Saturday	14	18	18	15	20	17
Sunday	15	19	19	16	21	18
Monday	16	20	20	17	22	19
Tuesday	17	21	21	18	23	20
Wednesday	18	22	22	19	24	21
Thursday	19	23	23	20	25	22
Friday	20	24	24	21	26	23
Saturday	21	25	25	22	27	24
Sunday	22	26	26	23	28	25
Monday	23	27	27	24	29	26
Tuesday	24	28	28	25	30	27
Wednesday	25		29	26	31	28
Thursday	26		30	27		29
Friday	27		31	28		30
Saturday	28			29		
Sunday	29			30		
Monday	30					
Tuesday	31					



ROOM MEASUREMENTS

1995

July	Aug	Sept	Oct	Nov	Dec	
						Monday
	1					Tuesday
	2			1		Wednesday
	3			2		Thursday
	4	1		3	1	Friday
	5	2		4	2	Saturday
	6	3	1	5	3	Sunday
	7	4	2	6	4	Monday
	8	5	3	7	5	Tuesday
	9	6	4	8	6	Wednesday
	10	7	5	9	7	Thursday
	11	8	6	10	8	Friday
	12	9	7	11	9	Saturday
	13	10	8	12	10	Sunday
0	14	11	9	13	11	Monday
1	15	12	10	14	12	Tuesday
2	16	13	11	15	13	Wednesday
3	17	14	12	16	14	Thursday
4	18	15	13	17	15	Friday
5	19	16	14	18	16	Saturday
6	20	17	15	19	17	Sunday
17	21	18	16	20	18	Monday
18	22	19	17	21	19	Tuesday
9	23	20	18	22	20	Wednesday
0	24	21	19	23	21	Thursday
1	25	22	20	24	22	Friday
2	26	23	21	25	23	Saturday
3	27	24	22	26	24	Sunday
4	28	25	23	27	25	Monday
5	29	26	24	28	26	Tuesday
6	30	27	25	29	27	Wednesday
7	31	28	26	30	28	Thursday
8		29	27		29	Friday
9		30	28		30	Saturday
0			29		31	Sunday
1			30			Monday
			31			Tuesday

Programming by Charlote Greenwood

Programming.
It's everywhere, so at an early age
after a spell with electronics
finding it too physical and here and past
found programming; meta-physical
in the here and after and before
depending; depends.

Creative female creations.
Perhaps one day will I be as flexible
as programs should.

Within this {now}

A7M

deadpay 1904